AWS IoT Device Defender 개발자 안내서

AWS IoT Device Defender

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS IoT Device Defender: AWS IoT Device Defender 개발자 안내서

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 관련하여 고객에게 혼동을 일으킬 수 있는 방식이나 Amazon 브랜드 이미지를 떨어뜨리는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS IoT Device Defender이란 무엇인가요? 1
AWS IoT Device Defender를 처음 사용하십니까? 2
AWS IoT Device Defender 작동 방식 2
AWS IoT Device Defender의 기능 3
AWS IoT Device Defender를 시작하는 방법 5
관련 서비스 5
AWS IoT Device Defender 액세스 5
AWS IoT Device Defender 요금 5
AWS IoT Device Defender 시작하기 6
설정 6
AWS 계정에 등록 6
관리자 액세스 권한이 있는 사용자 생성 7
감사 가이드 8
필수 조건 8
감사 검사 활성화 9
감사 결과 보기 9
감사 완화 작업 생성 9
감사 결과에 완화 작업 적용 10
AWS IoT Device Defender Audit IAM 역할 생성(선택 사항) 10
SNS 알림 활성화(선택 사항) 11
로깅 활성화(선택 사항) 12
ML Detect 가이드 12
사전 조건 13
콘솔에서 ML Detect를 사용하는 방법 13
CLI에서 ML 검색을 사용하는 방법 29
AWS IoT Device Defender 감사 결과를 보는 시기와 방법 사용자 지정 43
시작하기 44
콘솔에서 감사 결과 사용자 지정 44
CLI에서 감사 결과 사용자 지정 47
감사 55
문제 심각도 55
다음 단계 56
감사 검사 56
활성 디바이스 인증서에 대해 취소된 중간 CA 점검 57

취소된 CA 인증서가 계속 활성 상태 ... 58
공유된 디바이스 인증서 ... 59
디바이스 인증서 키 품질 ... 60
CA 인증서 키 품질 ... 62
인증되지 않은 Cognito 역할이 지나치게 허용됨 ... 64
인증된 Cognito 역할이 지나치게 허용됨 ... 71
지나치게 허용적인 AWS IoT 정책 ... 80
잘못 구성되었을 가능성이 있는 AWS IoT 정책 ... 86
역할 별칭이 지나치게 허용됨 ... 91
역할 별칭으로 사용되지 않는 서비스에 대한 액세스 허용 ... 92
CA 인증서 만료 ... 93
충돌하는 MQTT 클라이언트 ... 94
디바이스 인증서 만료 ... 95
디바이스 인증서 수명 확인 ... 97
취소된 디바이스 인증서가 계속 활성 상태 ... 98
로깅 비활성화 ... 99
감사 명령 ... 100
감사 설정 관리 ... 100
감사 예약 ... 106
온디맨드 감사 실행 ... 119
감사 인스턴스 관리 ... 121
감사 결과 점검 ... 130
감사 결과 금지 ... 139
감사 결과 금지 작동 방식 ... 139
콘솔에서 감사 검사 금지를 사용하는 방법 ... 140
CLI에서 감사 결과 금지를 사용하는 방법 ... 147
감사 결과 금지 API ... 149
감지 ... 150
등록되지 않은 디바이스의 동작 모니터링 ... 151
보안 사용 사례 ... 152
클라우드 측 사용 사례 ... 152
디바이스 측 사용 사례 ... 154
개념 ... 158
동작 ... 160
ML Detect ... 163
ML Detect의 사용 사례 ... 164

ML Detect 작동 방식 164
최소 요구 사항 164
제한 사항 165
경보에서 거짓 양성 및 기타 확인 상태 표시 166
지원되는 지표 166
서비스 할당량 167
ML Detect CLI 명령 167
ML Detect API 167
ML Detect 보안 프로파일 일시 중지 또는 삭제 168
사용자 지정 지표 169
콘솔에서 사용자 지정 지표를 사용하는 방법 169
CLI에서 사용자 지정 지표를 사용하는 방법 172
사용자 지정 지표 CLI 명령 176
사용자 지정 지표 API 176
Device-side metrics 177
전송된 바이트(`aws:all-bytes-out`) 177
수신된 바이트(`aws:all-bytes-in`) 178
수신 TCP 포트 개수(`aws:num-listening-tcp-ports`) 180
수신 UDP 포트 수`aws:num-listening-udp-ports`) 181
전송된 패킷(`aws:all-packets-out`) 183
수신된 패킷(`aws:all-packets-in`) 185
대상 IP(`aws:destination-ip-addresses`) 186
수신 TCP 포트(`aws:listening-tcp-ports`) 187
수신 UDP 포트(`aws:listening-udp-ports`) 188
설정된 TCP 연결 수(`aws:num-established-tcp-connections`) 188
디바이스 지표 문서 사양 190
디바이스에서 지표 전송 198
클라우드 측 지표 199
메시지 크기(aws:message-byte-size) 199
전송된 메시지(aws:num-messages-sent) 200
수신된 메시지(aws:num-messages-received) 202
권한 부여 실패(aws:num-authorization-failures) 204
소스 IP(aws:source-ip-address) 205
연결 시도(aws:num-connection-attempts) 206
연결 해제(aws:num-disconnects) 207
연결 해제 기간(aws:disconnect-duration) 209

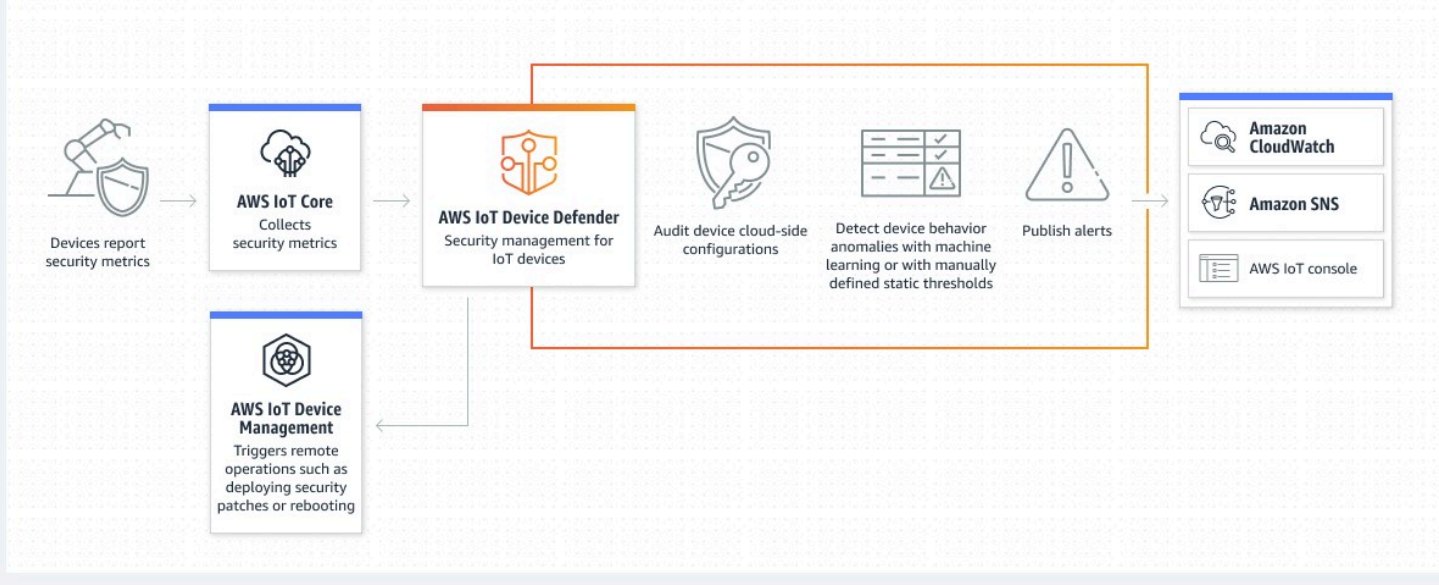
Detect 지표 내보내기 ... 210
지표 내보내기 탐지 작동 방식 ... 212
지표 내보내기 스키마 ... 212
지표 내보내기 탐지 요금 ... 213
권한 ... 214
AWS IoT 콘솔에서 지표 내보내기 탐지 설정 ... 215
보안 프로필을 생성하여 지표 내보내기 활성화 ... 217
보안 프로필을 업데이트하여 지표 내보내기 활성화(CLI) ... 218
보안 프로필을 업데이트하여 지표 내보내기 비활성화(CLI) ... 219
지표 내보내기 CLI 명령 ... 221
지표 내보내기 API 작업 ... 221
차원을 사용하여 보안 프로파일의 지표 범위 지정 ... 221
콘솔에서 차원을 사용하는 방법 ... 222
AWS CLI에서 차원을 사용하는 방법 ... 223
권한 ... 227
AWS IoT Device Defender Detect에 SNS 주제에 경보를 게시할 권한을 부여합니다. ... 228
Detect 명령 ... 229
AWS IoT Device Defender detect 사용 방법 ... 231
완화 작업 ... 234
감사 완화 작업 ... 234
완화 작업 감지 ... 238
완화 작업을 정의하고 관리하는 방법 ... 238
완화 작업 생성 ... 238
완화 작업 적용 ... 240
권한 ... 246
완화 작업 명령 ... 251
다른 AWS 서비스와 함께 AWS IoT Device Defender 사용 ... 252
AWS IoT Greengrass을(를) 실행하는 디바이스와 함께 AWS IoT Device Defender 사용 ... 252
FreeRTOS 및 임베디드 디바이스와 함께 AWS IoT Device Defender 사용 ... 252
AWS IoT Device Management와 함께 AWS IoT Device Defender 사용 ... 253
Security Hub 통합 ... 253
통합 활성화 및 구성 ... 254
AWS IoT Device Defender에서 Security Hub로 결과를 보내는 방법 ... 254
AWS IoT Device Defender의 일반적 결과 ... 256
AWS IoT Device Defender에서 결과를 Security Hub로 전송하는 작업 중지 ... 261
교차 서비스 혼동된 대리인 방지 ... 262

디바이스 에이전트의 보안 모범 사례 263
AWS IoT Device Defender 문제 해결 안내서 266
보안 271
데이터 보호 271
Identity and Access Management(IAM) 272
고객 273
ID를 통한 인증 273
정책을 사용한 액세스 관리 276
AWS IoT Device Defender에서 IAM을 사용하는 방식 279
자격 증명 기반 정책 예시 285
문제 해결 287
규정 준수 확인 289
복원력 290
문서 이력 291

AWS IoT Device Defender이란 무엇인가요?

디바이스 구성을 감사하고, 연결된 디바이스를 모니터링하고, 보안 위험을 완화하는 보안 및 모니터링 서비스인 AWS IoT Device Defender를 사용합니다. AWS IoT Device Defender를 사용하면 AWS IoT 디바이스 플릿 전반에서 일관성 있는 보안 정책을 적용하고, 디바이스가 손상된 경우 신속하게 대응할 수 있습니다. IoT 플릿은 다양한 기능을 수행하고 장기적으로 사용되며 지리적으로 분산된 다수의 디바이스로 구성될 수 있습니다. 이러한 특성으로 인해 플릿 설정이 복잡해지고 오류가 발생하기 쉬워집니다. 디바이스의 컴퓨팅 파워, 메모리 및 스토리지 기능이 한정된 경우가 많으므로 이에 따라 디바이스 자체에서 암호화 및 다른 형태의 보안을 사용하는 데 제약이 있습니다.

디바이스는 종종 취약점이 알려진 소프트웨어를 사용합니다. 이러한 요인으로 인해 IoT 플릿은 해커에게 매력적인 표적이며 디바이스 플릿을 지속적으로 보호하기가 어렵게 됩니다. AWS IoT Device Defender는 보안 문제와 모범 사례 위반을 식별하기 위한 도구를 제공하여 이러한 문제를 해결합니다. AWS IoT Device Defender는 디바이스 플릿 감사를 통해 플릿이 보안 모범 사례를 따르는지 확인하고 디바이스의 이상 동작을 감지할 수 있습니다. 다음 다이어그램은 AWS IoT Device Defender의 기본 아키텍처와 해당 제품이 AWS IoT Core, Amazon CloudWatch, Amazon SNS와 같은 서비스와 어떻게 연관되는지 보여 줍니다.



주제

- AWS IoT Device Defender를 처음 사용하십니까?
- AWS IoT Device Defender 작동 방식
- AWS IoT Device Defender의 기능
- AWS IoT Device Defender를 시작하는 방법

- 관련 서비스
- AWS IoT Device Defender 액세스
- AWS IoT Device Defender 요금

AWS IoT Device Defender를 처음 사용하십니까?

AWS IoT Device Defender을(를) 처음 사용할 경우 먼저 다음의 섹션을 읽는 것이 좋습니다.

- AWS IoT Device Defender 작동 방식
- AWS IoT Device Defender의 기능
- AWS IoT Device Defender를 시작하는 방법
- 관련 서비스
- AWS IoT Device Defender 액세스
- AWS IoT Device Defender 요금

AWS IoT Device Defender 작동 방식

AWS IoT Device Defender는 IoT 디바이스 플릿을 보호하는 데 도움이 되는 완전관리형 보안 및 모니터링 서비스입니다. AWS IoT Device Defender는 디바이스와 관련된 IoT 리소스를 감사하여 보안 모범 사례를 준수하는지 확인합니다. 감사 검사는 감지된 보안 위험이 있는 경우 경고를 보내고 문제를 완화하는 데 도움이 되는 관련 정보를 제공합니다. AWS IoT Device Defender는 또한 클라우드와 디바이스 측의 보안 지표를 지속적으로 모니터링함으로써 예상치 못한 디바이스 동작을 감지하여 손상 가능성이 있는 디바이스를 식별합니다. 필요에 따라 또는 일정에 따라 감사 검사를 시작하여 IoT 디바이스 구성을 평가할 수 있습니다.

AWS IoT Device Defender는 AWS IoT Core와 연동되어 디바이스 상호 작용의 컨텍스트를 통합함으로써 감사 검사의 정확성을 높입니다. AWS IoT Device Defender는 연결된 디바이스에서 중요한 보안 지표를 수집 및 분석하여 비정상적인 동작을 감지합니다. Rules Detect를 사용하면 사용자가 정의한 동작을 기준으로 지표 데이터를 지속적으로 평가됩니다. ML Detect를 사용하면 이상 징후를 식별하기 위해 자동으로 구축된 기계 학습(ML) 모델을 통해 지표 데이터가 지속적으로 평가됩니다.

예정된 감사 작업의 결과와 감지된 모든 디바이스 활동 이상이 AWS IoT Console 및 AWS IoT Device Defender API에 게시됩니다. Amazon CloudWatch를 통해 액세스할 수 있습니다. 또한 AWS IoT Device Defender가 결과를 Amazon SNS 주제로 전송하여 보안 대시보드와 통합하거나 자동 문제 해결 워크플로를 시작하도록 구성할 수 있습니다.

AWS IoT Device Defender는 다음을 비롯한 여러 사용 사례를 지원합니다.

- 디바이스 보호: AWS IoT 보안 모범 사례에 따라 디바이스 관련 리소스를 감사하여 디바이스 취약성을 감지하는 데 도움이 될 수 있습니다. AWS IoT Device Defender 감사를 통해 디바이스에 대한 위험을 식별 및 발견하고 보안 조치가 마련되어 있는지 확인할 수 있습니다.
- 비정상적인 디바이스 동작 감지: 연결 패턴의 변화를 정확히 확인하고, 인증되지 않은 엔드포인트와의 디바이스 통신을 찾아내고, 인바운드 및 아웃바운드 디바이스 트래픽 패턴의 변화를 식별할 수 있습니다.
- 위험 완화를 위한 인사이트 확보: 감사 조사 결과 또는 감지 경보에서 발견된 문제를 완화하기 위한 조치를 취할 수 있습니다.
- 디바이스 보안 유지: 감사 및 감지 검사에서 얻은 인사이트를 사용하여 발생 가능한 보안 침해를 진단하고 해결할 수 있습니다.
- 디바이스 보안 강화: 잘못 구성된 디바이스를 구별하고, 디바이스 플릿의 상태를 조사하고, 예상치 못한 디바이스 동작 지표를 찾을 수 있습니다.

AWS IoT Device Defender의 기능

다음은 AWS IoT Device Defender의 몇 가지 주요 기능입니다.

주요 기능

감사	AWS IoT Device Defender는 IAM 사용 설명서의 AWS IoT 보안 모범 사례에 따라 디바이스 관련 리소스를 감사합니다. AWS IoT Device Defender는 보안 모범 사례를 준수하지 않는 구성을 보고합니다. 한 디바이스에서 다른 여러 디바이스의 데이터를 읽고 업데이트할 수 있도록 허용하는 지나치게 관대한 정책을 이러한 구성의 예로 들 수 있습니다.
Rules Detect	AWS IoT Device Defender는 디바이스 및 AWS IoT Core에서 중요한 보안 지표를 지속적으로 모니터링하여 보안 침해를 나타낼 수 있는 비정상적인 디바이스 동작을 감지합니다. 이러한 지표에 대한 동작(규칙)을 설정하여 디바이스 그룹

	의 정상적인 디바이스 동작을 지정할 수 있습니다. AWS IoT Device Defender는 이러한 지표에 대해 보고된 각 데이터 포인트를 사용자가 정의한 동작(규칙)과 비교하여 모니터링 및 평가하고 이상이 감지되면 경고합니다.
ML Detect	AWS IoT Device Defender는 지난 14일 기간의 클라우드 측 지표 6개와 디바이스 측 지표 7개에 대한 디바이스 데이터를 사용하여 기계 학습(ML) 모델을 통해 자동으로 디바이스 동작을 설정합니다. 그런 다음 모델을 학습시킬 수 있는 충분한 데이터가 있는 한, 매일 모델을 재학습시켜 초기 모델이 구축된 후의 최근 14일을 기준으로 예상 디바이스 동작을 새로 고칩니다. AWS IoT Device Defender는 ML 모델을 사용하여 이러한 지표에 대한 이상 데이터 포인트를 모니터링 및 식별하고, 이상이 감지되면 경보를 트리거합니다.
알림	AWS IoT Device Defender는 AWS IoT 콘솔, Amazon CloudWatch, Amazon SNS에 경보를 게시합니다.
완화	AWS IoT Device Defender는 디바이스 메타데이터, 디바이스 통계, 디바이스에 대한 기간별 알림 등 디바이스에 대한 상황별 및 기간별 정보를 제공하여 문제를 조사하는 데 사용할 수 있습니다. 또한 AWS IoT Device Defender에 내장된 완화 작업을 사용하여 사물 그룹에 항목 추가, 기본 정책 버전 교체, 디바이스 인증서 업데이트와 같은 감사 및 감지 경보에 대한 완화 단계를 수행할 수 있습니다.

AWS IoT Device Defender를 시작하는 방법

AWS IoT Device Defender를 시작하는 데 도움이 필요하면 다음 자습서를 참조하세요.

- 설정
- ML Detect 가이드
- 감사 가이드
- AWS IoT Device Defender 감사 결과를 보는 시기와 방법 사용자 지정

관련 서비스

- AWS IoT Greengrass: AWS IoT Greengrass는 AWS IoT Device Defender와의 미리 빌드된 통합을 제공해 지속적으로 디바이스 동작을 모니터링합니다.
- AWS IoT Device Management: AWS IoT Device Management 플릿 인덱싱을 사용하여 AWS IoT Device Defender에서 감지된 위반을 인덱싱, 검색 및 집계할 수 있습니다.

AWS IoT Device Defender 액세스

AWS IoT Device Defender 콘솔 또는 API를 사용하여 AWS IoT Device Defender에 액세스할 수 있습니다.

AWS IoT Device Defender 요금

AWS IoT Device Defender는 사용한 만큼만 비용을 지불하면 됩니다. 최소 요금이나 필수 서비스 사용은 없습니다. 하지만 감사 및 감지 기능에 대해서는 별도로 요금이 청구됩니다. 감사 요금은 디바이스 개수당 월별로 계산됩니다. 감사를 켜면 한 달간의 활성 디바이스 보안 주체 수를 기준으로 요금이 청구됩니다. 따라서 감사 검사를 추가하거나 제거해도 이 기능을 사용할 때 월별 청구서에 영향을 미치지 않습니다. AWS 가격 계산기를 사용하면 단일 견적으로 AWS IoT Device Defender 및 아키텍처 비용을 계산할 수 있습니다.

- AWS 요금 계산기

AWS IoT Device Defender 시작하기

다음 자습서를 사용하여 AWS IoT Device Defender 작업을 수행할 수 있습니다.

주제

- 설정
- 감사 가이드
- ML Detect 가이드
- AWS IoT Device Defender 감사 결과를 보는 시기와 방법 사용자 지정

설정

AWS IoT Device Defender를 처음 사용한다면 먼저 다음 작업을 완료해야 합니다.

주제

- AWS 계정에 등록
- 관리자 액세스 권한이 있는 사용자 생성

AWS 계정에 등록

AWS 계정이 없는 경우 다음 절차에 따라 계정을 생성합니다.

AWS 계정에 가입

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따릅니다.

   등록 절차 중 전화 또는 텍스트 메시지를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   AWS 계정 루트 사용자에 가입하면 AWS 계정 루트 사용자가 만들어집니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS는 가입 절차 완료된 후 사용자에게 확인 이메일을 전송합니다. 언제든지 https://aws.amazon.com/으로 이동하고 내 계정을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

AWS 계정에 가입하고 AWS 계정 루트 사용자에 보안 조치를 한 다음, AWS IAM Identity Center을 활성화하고 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 생성합니다.

귀하의 AWS 계정 루트 사용자 보호

1. 루트 사용자를 선택하고 AWS 계정이메일 주소를 입력하여 AWS Management Console에 계정 소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 User Guide의 루트 사용자로 로그인을 참조하세요.
2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자용 가상 MFA 디바이스 활성화(콘솔)를 참조하세요.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center 설정을 참조하세요.
2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

   IAM Identity Center 디렉터리를 ID 소스로 사용하는 방법에 대한 자습서는 AWS IAM Identity Center 사용 설명서의 기본 IAM Identity Center 디렉터리로 사용자 액세스 구성을 참조하세요.

관리 액세스 권한이 있는 사용자로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

  IAM Identity Center 사용자로 로그인하는 데 도움이 필요한 경우 AWS 로그인 사용 설명서의 AWS 액세스 포털에 로그인을 참조하세요.

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 Add groups를 참조하세요.

이러한 작업은 AWS 계정 및 계정에 대한 관리자 권한을 가진 사용자를 생성합니다.

감사 가이드

이 자습서에서는 반복 감사 구성, 경보 설정, 감사 결과 검토 및 감사 문제 완화에 대한 지침을 제공합니다.

주제

- 필수 조건
- 감사 검사 활성화
- 감사 결과 보기
- 감사 완화 작업 생성
- 감사 결과에 완화 작업 적용
- AWS IoT Device Defender Audit IAM 역할 생성(선택 사항)
- SNS 알림 활성화(선택 사항)
- 로깅 활성화(선택 사항)

필수 조건

이 튜토리얼을 완료하려면 다음이 필요합니다.

- AWS 계정. 이것이 없는 경우 설정을 참조하세요.

감사 검사 활성화

다음 절차에서는 계정 및 디바이스 설정과 정책을 검토하는 감사 검사와 보안 조치가 구현되어 있는지 확인합니다. 이 자습서에서는 모든 감사 검사를 사용하도록 지시하지만 원하는 검사를 선택할 수 있습니다.

감사 요금은 디바이스 개수당 월별로 계산됩니다(AWS IoT에 연결된 플릿 디바이스). 따라서 감사 검사를 추가하거나 제거해도 이 기능을 사용할 때 월별 청구서에 영향을 미치지 않습니다.

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안을 확장하고 인트로를 선택합니다.
2. AWS IoT 보안 감사 자동화를 선택합니다. 감사 점검이 자동으로 켜집니다.
3. 감사를 확장하고 설정을 선택하여 감사 점검을 확인합니다. 감사 점검 이름을 선택하여 감사 점검의 기능을 알아보세요. 감사 검사에 대한 자세한 내용은 감사 검사를 참조하세요.
4. (선택 사항) 사용하려는 역할이 이미 있는 경우 서비스 권한 관리를 선택하고 목록에서 역할을 선택한 다음 업데이트를 선택합니다.

감사 결과 보기

다음 절차에서는 감사 결과를 보는 방법을 보여줍니다. 이 자습서에서는 감사 검사 활성화 자습서에서 설정한 감사 검사의 감사 결과를 봅니다.

감사 결과를 보려면

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 감사를 확장한 다음 결과를 선택합니다.
2. 조사할 감사 일정의 이름을 선택합니다.
3. 규정 미준수 점검의 완화에서 정보 버튼을 선택하여 규정 미준수 사유에 대한 정보를 확인하세요. 규정 미준수 점검이 규정을 준수하도록 하는 방법에 대한 지침은 감사 검사 섹션을 참조하세요.

감사 완화 작업 생성

다음 절차에서는 AWS IoT 로깅을 활성화할 AWS IoT Device Defender 감사 완화 작업을 생성합니다. 각 감사 검사에는 매핑된 완화 작업이 있으며 이러한 작업은 수정하려는 감사 검사에 대해 선택하는 작업 유형에 영향을 줍니다. 자세한 내용은 완화 작업을 참조하세요.

AWS IoT 콘솔을 사용하여 완화 작업을 생성하려면

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 탐지를 확장한 다음 완화 작업을 선택합니다.

2. Mitigation actions(완화 작업) 페이지에서 Create(생성)를 선택합니다.
3. 새 완화 작업 생성 페이지의 작업 이름에서 완화 작업에 고유한 이름(예: *EnableErrorLoggingAction*)을 입력합니다.
4. 작업 유형에서 AWS IoT 로깅 활성화를 선택합니다.
5. 권한에서 역할 생성을 선택합니다. 역할 이름에 *IoTMitigationActionErrorLoggingRole*을 사용합니다. 그다음에 생성을 선택합니다.
6. 파라미터에서 로깅을 위한 역할에 `IoTMitigationActionErrorLoggingRole`을 선택합니다. 로그 수준에 `Error`을(를) 선택합니다.
7. 생성(Create)을 선택합니다.

감사 결과에 완화 작업 적용

다음 절차에서는 감사 결과에 완화 작업을 적용하는 방법을 소개합니다.

비준수 감사 결과를 완화하려면

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 감사를 확장한 다음 결과를 선택합니다.
2. 대응하려는 감사 결과를 선택합니다.
3. 결과를 확인하세요.
4. Start mitigation actions(완화 작업 시작)를 선택합니다.
5. 로깅이 비활성화된 경우 이전에 만든 완화 작업인 `EnableErrorLoggingAction`을 선택합니다. 각 규정 미준수 결과에 대해 적절한 작업을 선택하여 문제를 해결합니다.
6. 사유 코드 선택에서 감사 점검에서 반환된 사유 코드를 선택합니다.
7. 작업 시작을 선택합니다. 완화 작업을 실행하는 데 몇 분이 걸릴 수 있습니다.

완화 작업이 작동하는지 확인하려면

1. AWS IoT 콘솔의 탐색 창에서 설정을 선택합니다.
2. 서비스 로그에서 로그 수준이 `Error (least verbosity)`인지 확인합니다.

AWS IoT Device Defender Audit IAM 역할 생성(선택 사항)

다음 절차에서 AWS IoT에 대한 읽기 액세스 권한을 AWS IoT Device Defender에 제공하는 AWS IoT Device Defender Audit IAM 역할을 생성합니다.

AWS IoT Device Defender에 대한 서비스 역할을 생성하는 방법 (콘솔)

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다.
3. AWS 서비스 역할 유형을 선택합니다.
4. 다른 AWS 서비스의 사용 사례에서 AWS IoT를 선택한 다음 IoT - Device Defender Audit를 선택합니다.
5. 다음을 선택합니다.
6. (선택 사항) 권한 경계를 선택합니다. 이는 서비스 역할에서 가능한 고급 기능이며 서비스 링크된 역할은 아닙니다.

   권한 경계 섹션을 열고 최대 역할 권한을 관리하기 위한 권한 경계 사용을 선택합니다. IAM은 계정의 AWS 관리형 또는 고객 관리형 정책 목록을 포함합니다. 권한 경계를 사용하기 위한 정책을 선택하거나 정책 생성을 선택하여 새 브라우저 탭을 열고 완전히 새로운 정책을 생성합니다. 자세한 내용은 IAM 사용자 설명서에서 IAM 정책 생성을 참조하세요. 정책을 생성하면 탭을 닫고 원래 탭으로 돌아와 권한 경계에 사용할 정책을 선택합니다.
7. 다음을 선택합니다.
8. 이 역할의 목적을 식별하는 데 도움이 되는 역할 이름을 입력합니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 **PRODROLE**과 **prodrole**, 두 가지로 지정된 역할을 만들 수는 없습니다. 다양한 개체가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
9. (선택 사항)설명에 새 역할에 대한 설명을 입력합니다.
10. 1단계: 신뢰할 수 있는 엔터티 선택(Step 1: Select trusted entities) 또는 2단계: 권한 선택(Step 2: Select permissions) 섹션에서 편집(Edit)을 선택하여 역할에 대한 사용 사례와 권한을 편집합니다.
11. (선택 사항) 태그를 키 값 페어로 연결하여 메타데이터를 사용자에게 추가합니다. IAM에서 태그 사용에 대한 자세한 내용을 알아보려면 IAM 사용 설명서의 IAM 리소스에 태그 지정을 참조하십시오.
12. 역할을 검토한 다음 역할 생성을 선택합니다.

SNS 알림 활성화(선택 사항)

다음 절차에서는 Amazon SNS(SNS) 알림을 사용하여 감사에서 규정 미준수 리소스를 식별할 때 알림을 받습니다. 이 자습서에서는 감사 검사 활성화 자습서에서 활성화된 감사 검사에 대한 알림을 설정합니다.

1. 아직 연결하지 않았다면 AWS Management Console을 통해 SNS에 액세스할 수 있는 정책을 연결하세요. IAM 사용 설명서의 정책을 IAM 사용자 그룹에 연결의 지침을 따르고 AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction 정책을 선택하면 됩니다.
2. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 감사를 확장한 다음 설정을 선택합니다.
3. Device Defender 감사 설정 페이지 하단에서 SNS 알림 활성화를 선택합니다.
4. 활성을 선택합니다.
5. 주제에서 새 주제 생성을 선택합니다. 주제 이름을 *IoTDDNotifications*로 지정하고 생성을 선택합니다. 역할에서 AWS IoT Device Defender Audit IAM 역할 생성(선택 사항)에서 생성한 역할을 선택합니다.
6. 업데이트를 선택합니다.
7. Amazon SNS를 통해 Ops 플랫폼에서 이메일이나 문자를 수신하려면 사용자 알림에 Amazon Simple Notification Service 사용을 참조하세요.

로깅 활성화(선택 사항)

이 절차에서는 AWS IoT이(가) CloudWatch Logs에 정보를 로깅하도록 설정하는 방법을 설명합니다. 이렇게 하면 감사 결과를 볼 수 있습니다. 로깅을 활성화하면 요금이 발생할 수 있습니다.

로깅을 활성화하려면

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 설정을 선택합니다.
2. 로그에서 로그 관리를 선택합니다.
3. 역할 선택에서 역할 생성을 선택합니다. 역할 이름을 *AWSIoTLoggingRole*로 지정하고 생성을 선택합니다. 정책이 자동으로 연결됩니다.
4. 로그 수준에서 디버그(최대 상세 수준)를 선택합니다.
5. 업데이트를 선택합니다.

ML Detect 가이드

이 시작 안내서에서는 기계 학습(ML)을 사용하여 디바이스의 기록 지표 데이터를 기반으로 예상되는 동작의 모델을 만드는 ML Detect 보안 프로파일을 만듭니다. ML Detect가 ML 모델을 생성하는 동안 진행 상황을 모니터링할 수 있습니다. ML 모델을 빌드한 후 지속적으로 경보를 확인 및 조사하고 식별된 문제를 완화할 수 있습니다.

ML Detect 및 해당 API 및 CLI 명령에 대한 자세한 내용은 ML Detect 단원을 참조하세요.

이번 장은 다음과 같은 단원들로 구성되어 있습니다.

- 사전 조건
- 콘솔에서 ML Detect를 사용하는 방법
- CLI에서 ML 검색을 사용하는 방법

사전 조건

- AWS 계정. 이것이 없는 경우 설정을 참조하세요.

콘솔에서 ML Detect를 사용하는 방법

자습서

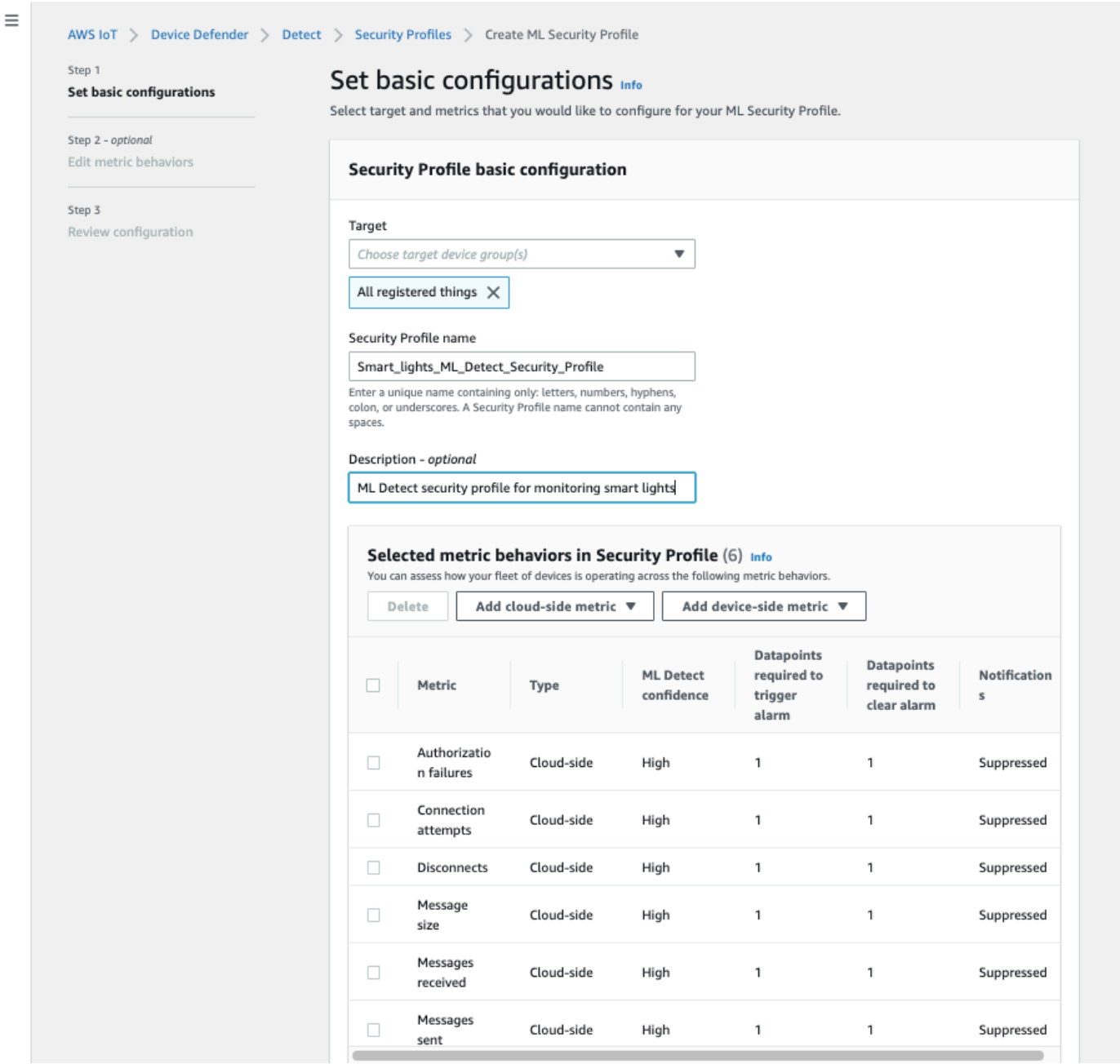
- ML Detect 활성화
- ML 모델 상태 모니터링
- ML Detect 경보 검토
- ML 경보 미세 조정
- 경보의 확인 상태 표시
- 식별된 디바이스 문제 완화

ML Detect 활성화

다음 절차에서는 콘솔에서 ML Detect를 설정하는 방법에 대해 소개합니다.

1. 먼저, 모델의 지속적인 교육 및 새로 고침에 대해 ML Detect 최소 요구 사항에 정의된 대로 디바이스에서 필요한 최소 데이터 포인트를 생성하는지 확인합니다. 데이터 수집을 진행하려면 보안 프로파일이 대상(사물 또는 사물 그룹)에 연결되어 있어야 합니다.
2. AWS IoT 콘솔의 탐색 창에서 방어를 확장합니다. 감지, 보안 프로파일, 보안 프로파일 생성, ML 예외 항목 감지 프로파일 만들기를 차례로 선택합니다.
3. 기본 구성 설정 페이지에서 다음 중 하나를 수행합니다.
   - 대상 아래에서 대상 디바이스 그룹을 선택합니다.
   - 보안 프로파일 이름 아래에서 보안 프로파일의 이름을 입력합니다.
   - (선택 사항) 설명 아래에서 ML 프로파일에 대한 간단한 설명을 작성할 수 있습니다.

- 보안 프로파일에서 선택한 지표 동작 아래에서 모니터링할 지표를 선택합니다.

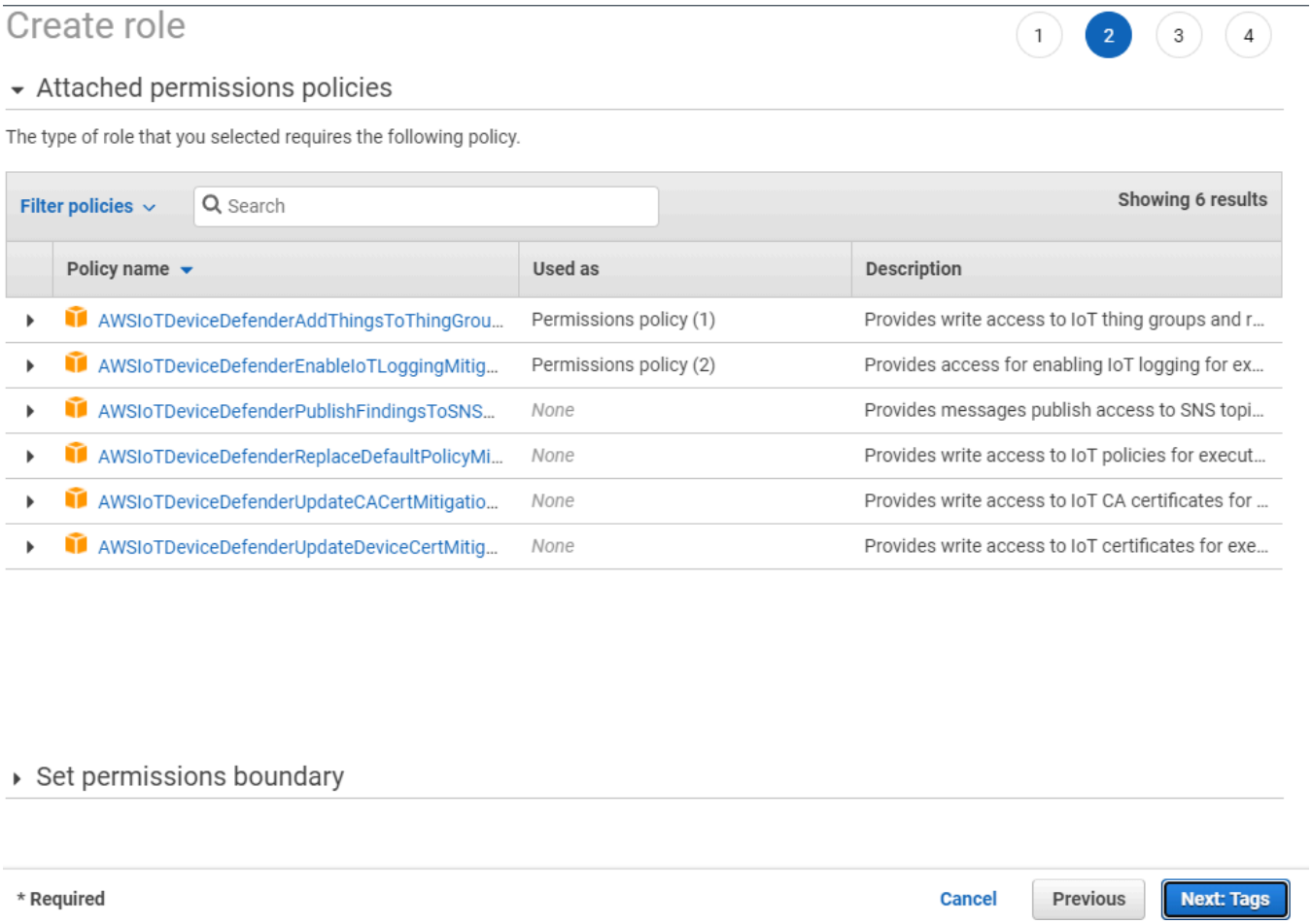


완료했으면 다음을 선택합니다.

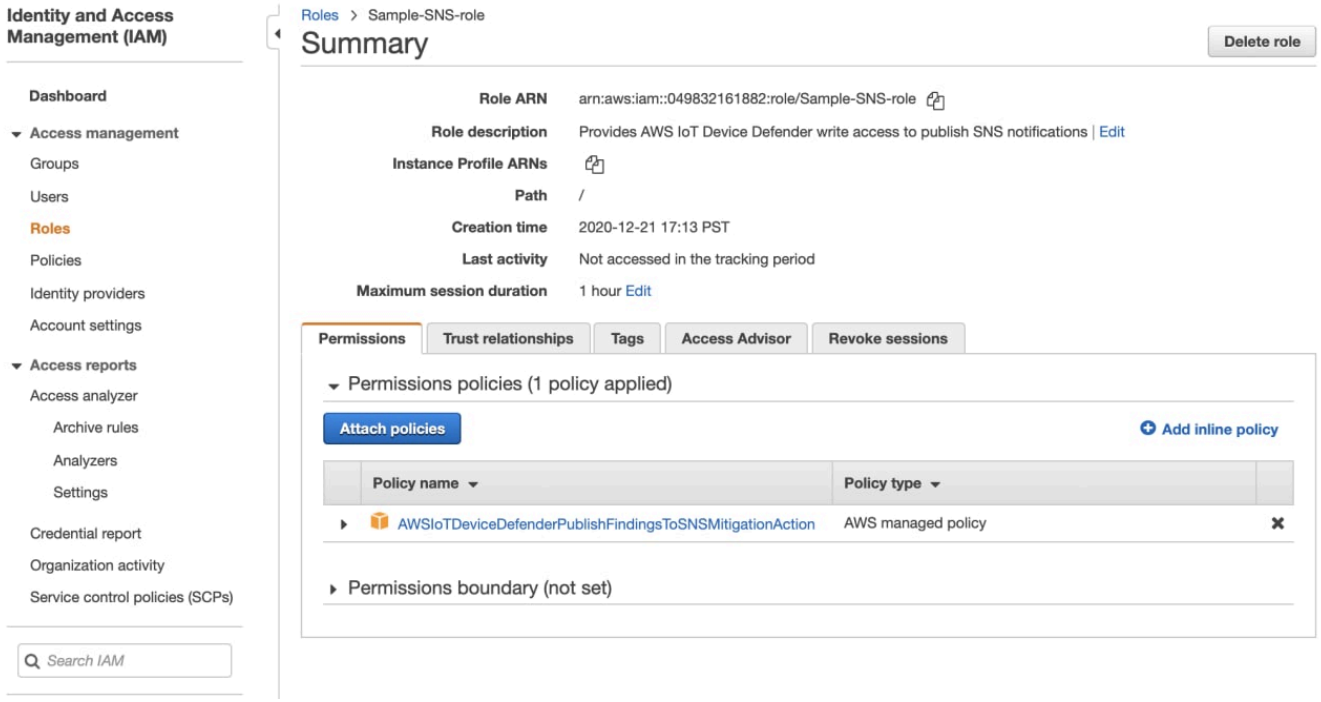
4. SNS 설정(선택 사항) 페이지에서 디바이스가 프로파일의 동작을 위반할 때 경보 알림에 대한 SNS 주제를 지정합니다. 선택한 SNS 주제에 게시할 때 사용할 IAM 역할을 선택합니다.

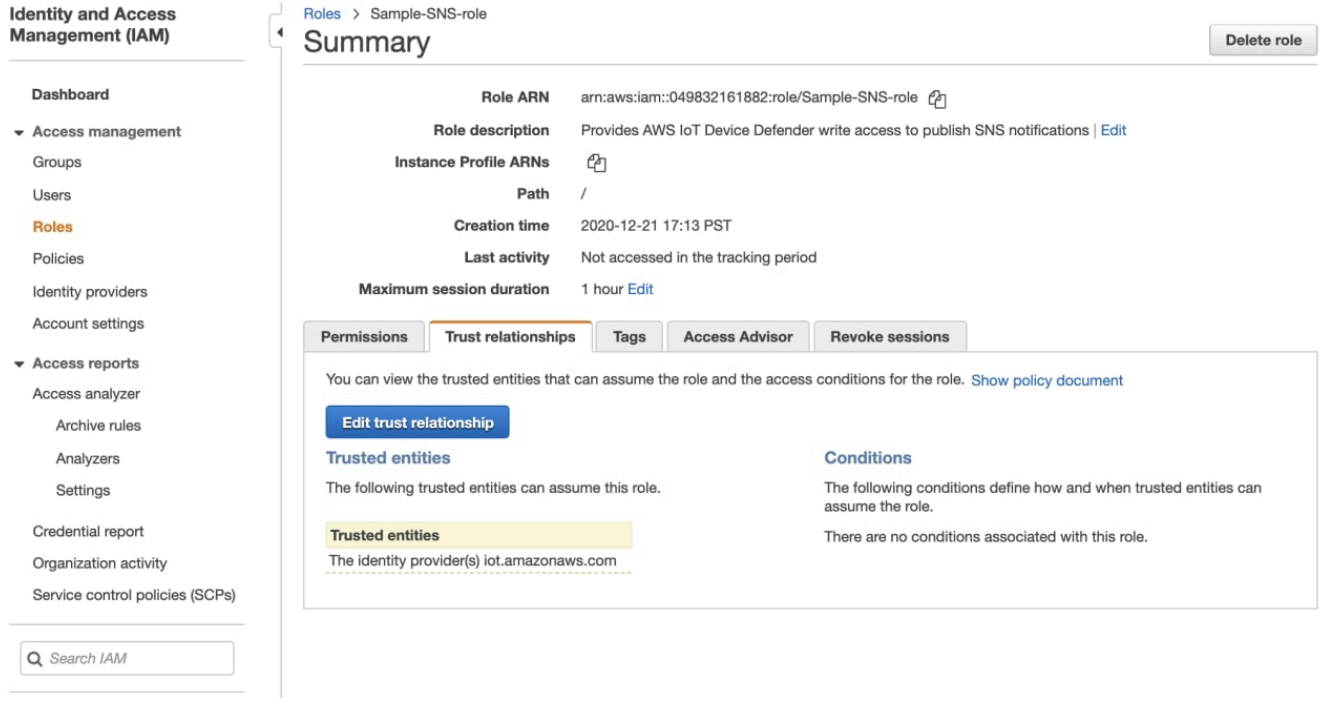
   아직 SNS 역할이 없는 경우 다음 단계에 따라 적절한 권한 및 신뢰 관계가 필요한 역할을 만듭니다.

- IAM 콘솔로 이동합니다. 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
- 신뢰할 수 있는 유형의 엔터티 선택에서 AWS 서비스를 선택합니다. 그런 다음 사용 사례 선택에서 IoT를 선택하고 사용 사례 선택에서 IoT - Device Defender 완화 작업을 선택합니다. 완료했으면 다음: 권한을 선택합니다.
- 연결된 권한 정책에서 AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction을 선택한 후 다음: 태그를 선택해야 합니다.

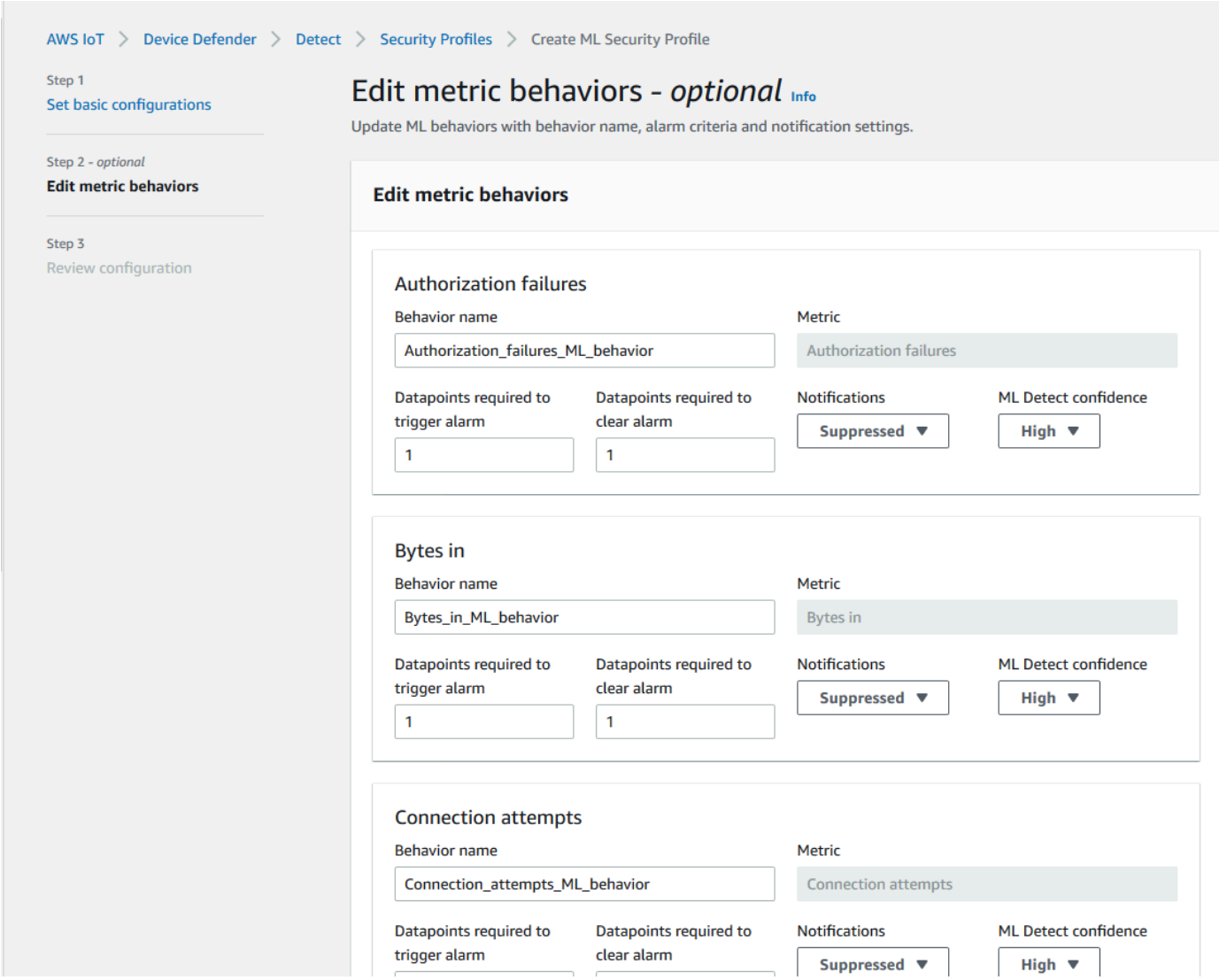


- 태그 추가(선택 사항)에서 역할과 연결하려는 태그를 추가할 수 있습니다. 완료했으면 Next: Review(다음: 검토)를 선택합니다.
- 검토에서 역할의 이름을 지정하고 AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction이 권한 아래에 나열되고 AWS 서비스: iot.amazonaws.com이 신뢰 관계 아래에 나열되어 있어야 합니다. 완료했으면 역할 생성을 선택하세요.

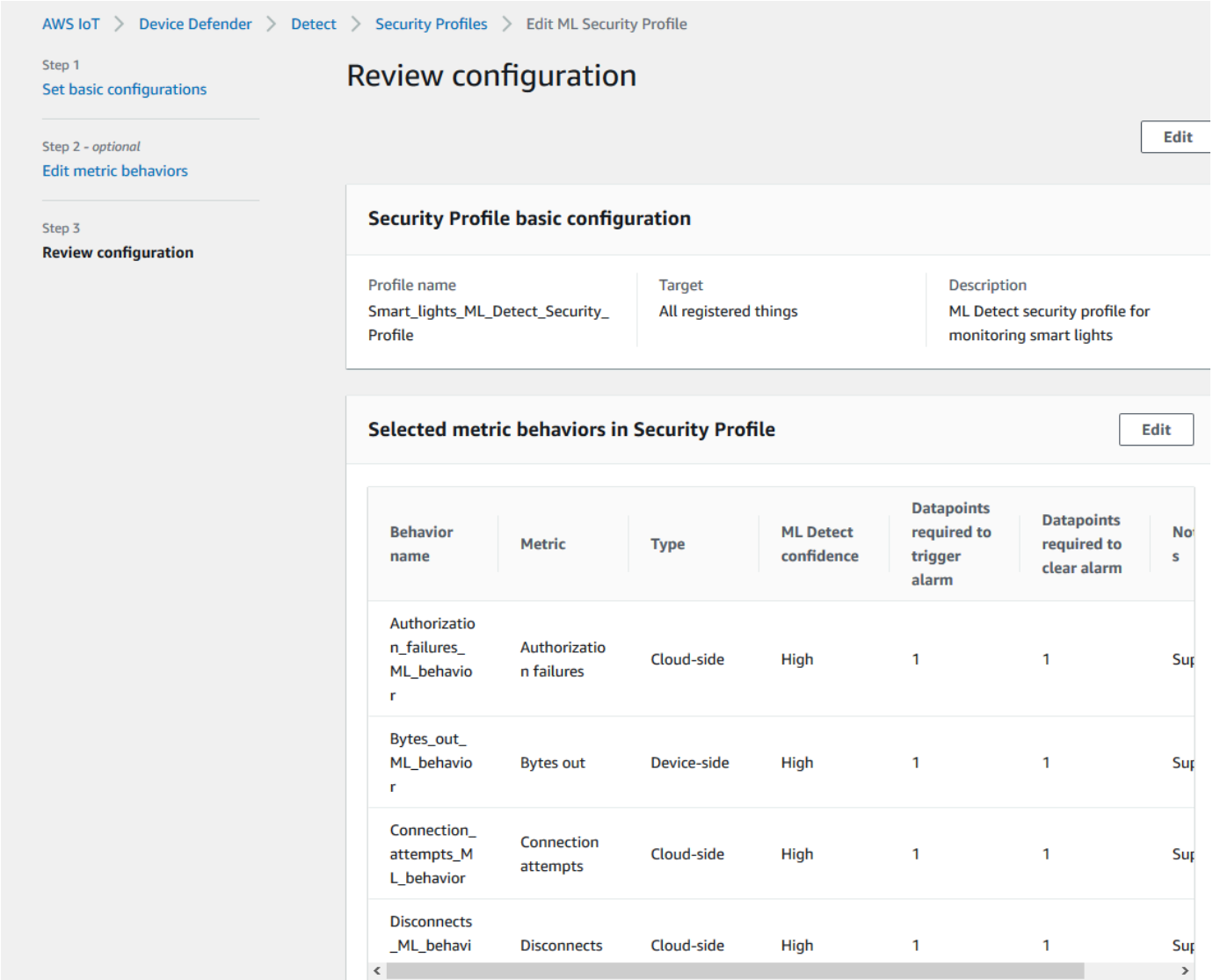




5. 지표 편집 동작 페이지에서 ML 동작 설정을 사용자 지정할 수 있습니다.



6. 완료했으면 다음을 선택합니다.
7. 구성 검토 페이지에서 기계 학습에서 모니터링할 동작을 확인하고 다음을 선택합니다.



8. 보안 프로파일을 생성한 후에는 보안 프로파일 페이지로 리디렉션되어 새로 만든 보안 프로파일이 나타납니다.

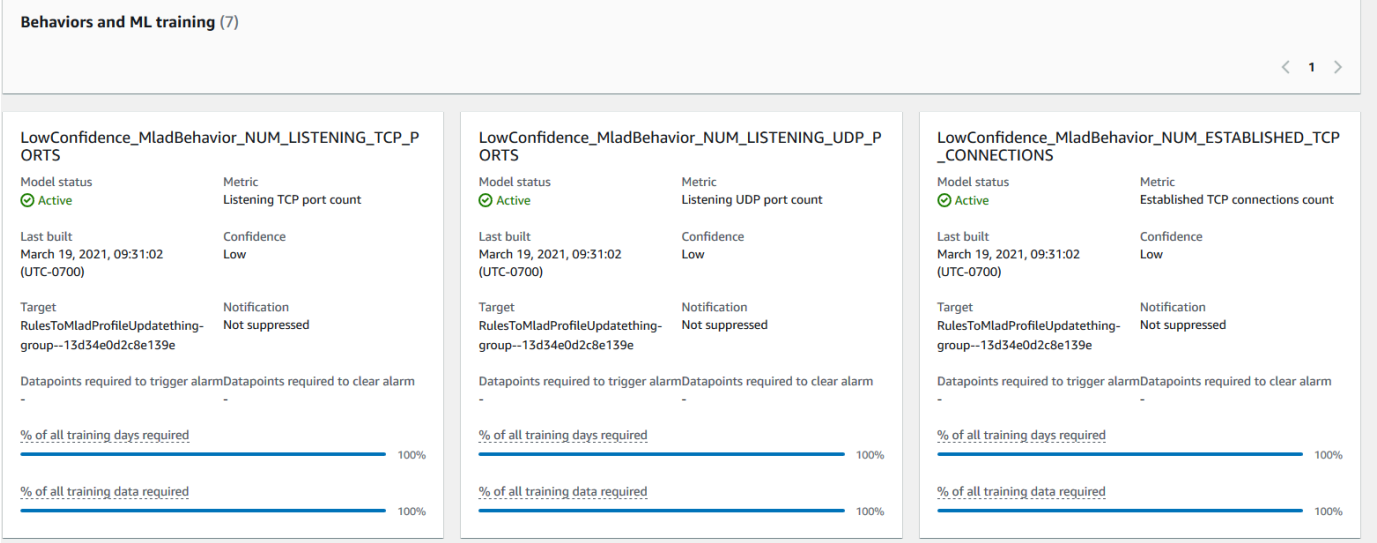
> **Note**
>
> 초기 ML 모델 훈련 및 제작을 완료하는 데 14일이 걸립니다. 디바이스에 비정상적인 활동이 있는 경우 완료된 후 경보가 표시될 것으로 예상할 수 있습니다.

ML 모델 상태 모니터링

ML 모델이 초기 교육 기간에 있는 동안 다음 단계를 수행하여 언제든지 진행 상황을 모니터링할 수 있습니다.

1. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 보안 프로파일을 차례로 선택합니다.
2. 보안 프로파일 페이지에서 검토하려는 보안 프로파일을 선택합니다. 그런 다음 동작 및 ML 교육을 선택합니다.
3. 동작 및 ML 교육 페이지에서 ML 모델의 교육 진행 상황을 확인하세요.

   모델 상태가 활성 상태이면 사용량에 따라 감지 결정을 내리고 매일 프로파일을 업데이트합니다.

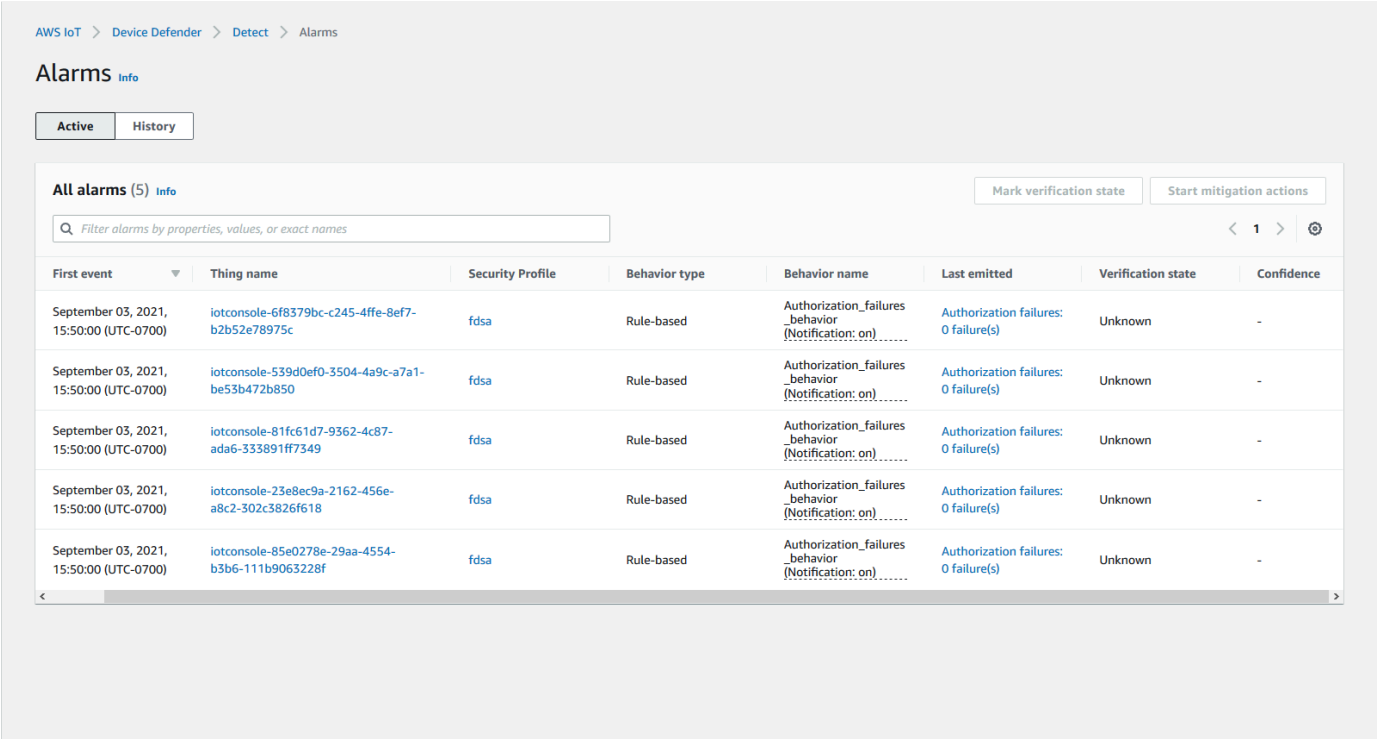


> **Note**
>
> 모델이 예상대로 진행되지 않는 경우 디바이스가 최소 요구 사항을 충족하는지 확인합니다.

ML Detect 경보 검토

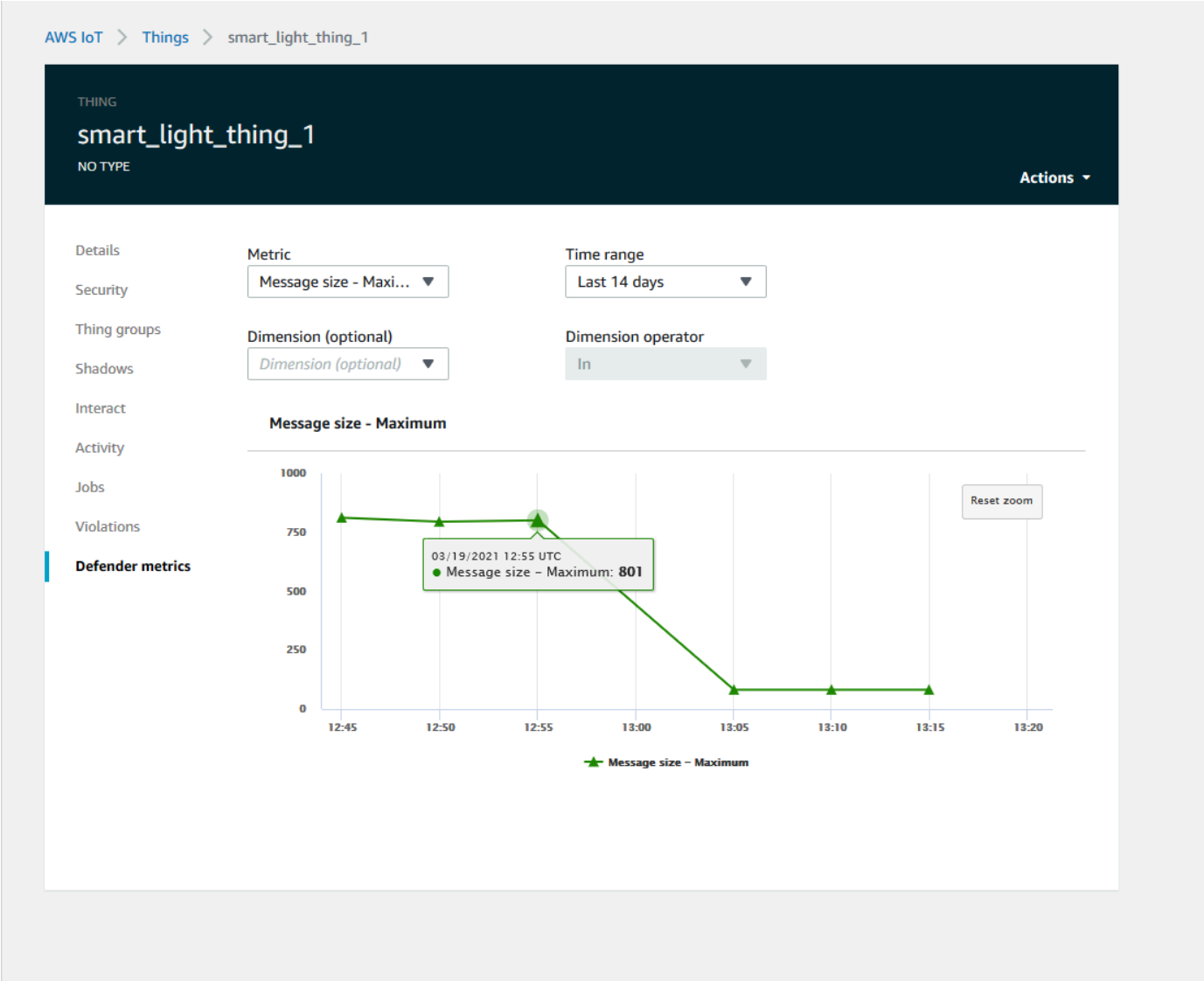
ML 모델을 빌드하고 데이터 추론을 준비한 후에는 모델에서 식별되는 경보를 정기적으로 확인 및 조사할 수 있습니다.

1. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 경보를 차례로 선택합니다.



2. 기록 탭으로 이동하여 더 이상 경보에 표시되지 않는 디바이스에 대한 세부 정보를 볼 수도 있습니다.

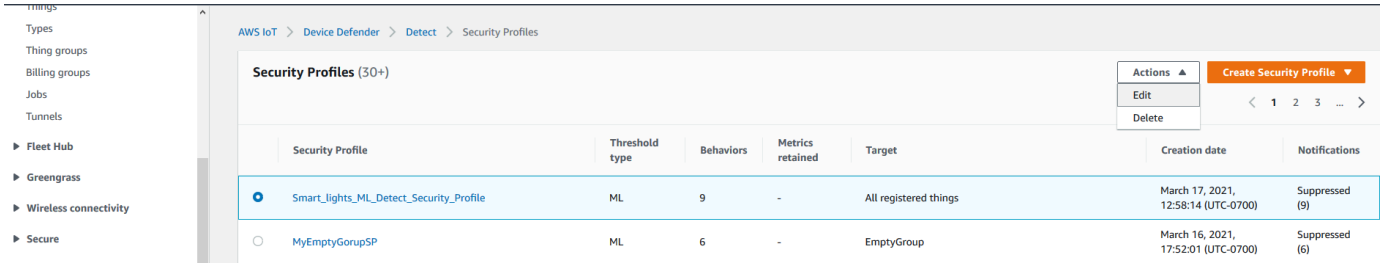
자세한 정보를 보려면 관리 아래 사물에서 자세한 내용을 보고 싶은 사물을 선택한 다음 Defender 지표로 이동합니다. Defender 지표 그래프에 액세스하고 활성 상태 탭에서 경보에 표시된 모든 것에 대한 조사를 수행할 수 있습니다. 이 경우 경보를 시작한 메시지 크기의 스파이크가 그래프에 표시됩니다. 이후에 경보가 지워지는 것을 볼 수 있습니다.



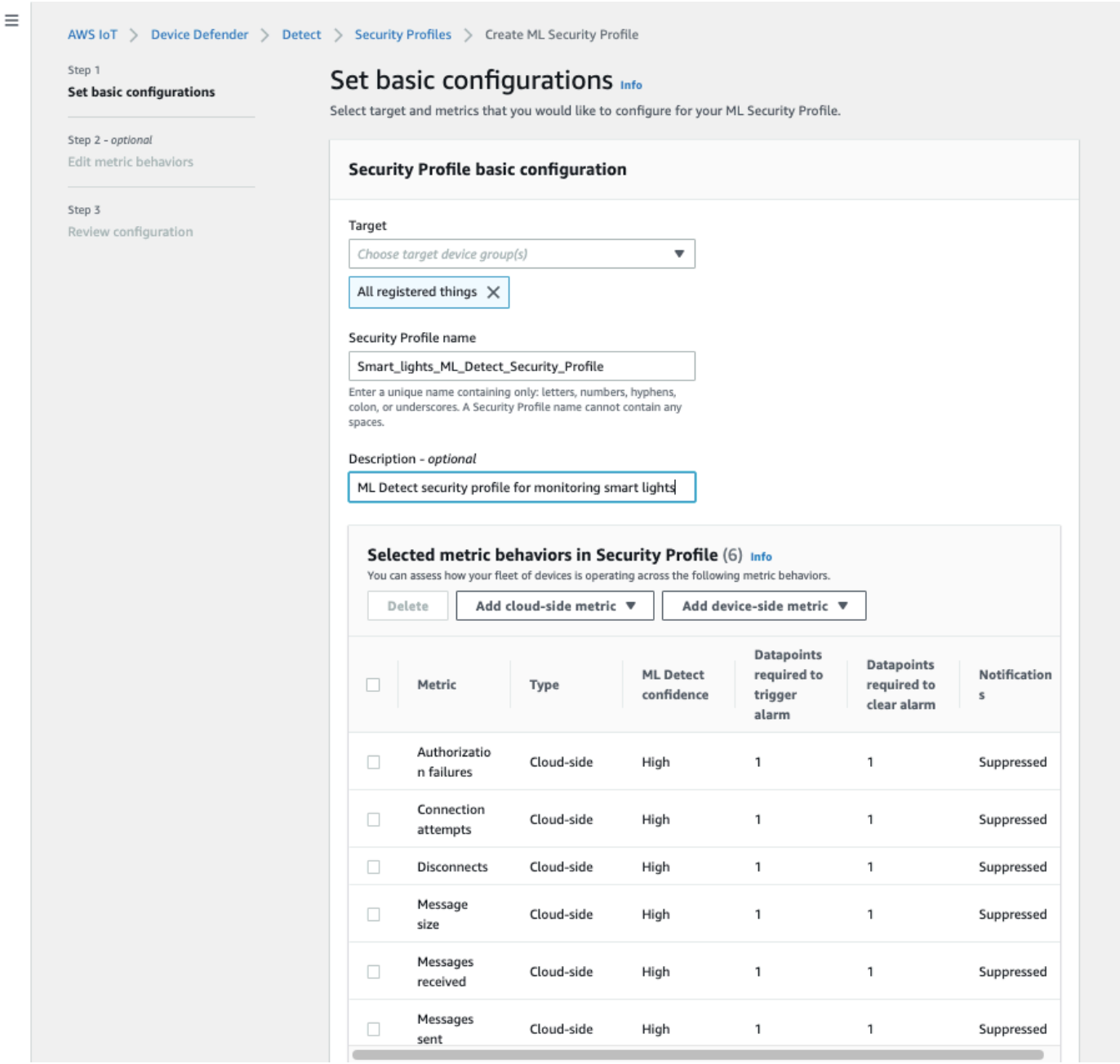
ML 경보 미세 조정

ML 모델을 빌드하고 데이터 평가를 위한 준비가 완료되면 보안 프로파일의 ML 동작 설정을 업데이트하여 구성을 변경할 수 있습니다. 다음 절차에서는 AWS CLI에서 보안 프로파일의 ML 동작 설정을 업데이트하는 방법을 보여 줍니다.

1. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 보안 프로파일을 차례로 선택합니다.
2. 보안 프로파일 페이지에서 검토하려는 보안 프로파일 옆에 있는 확인란을 선택합니다. 작업, 편집을 차례로 선택합니다.

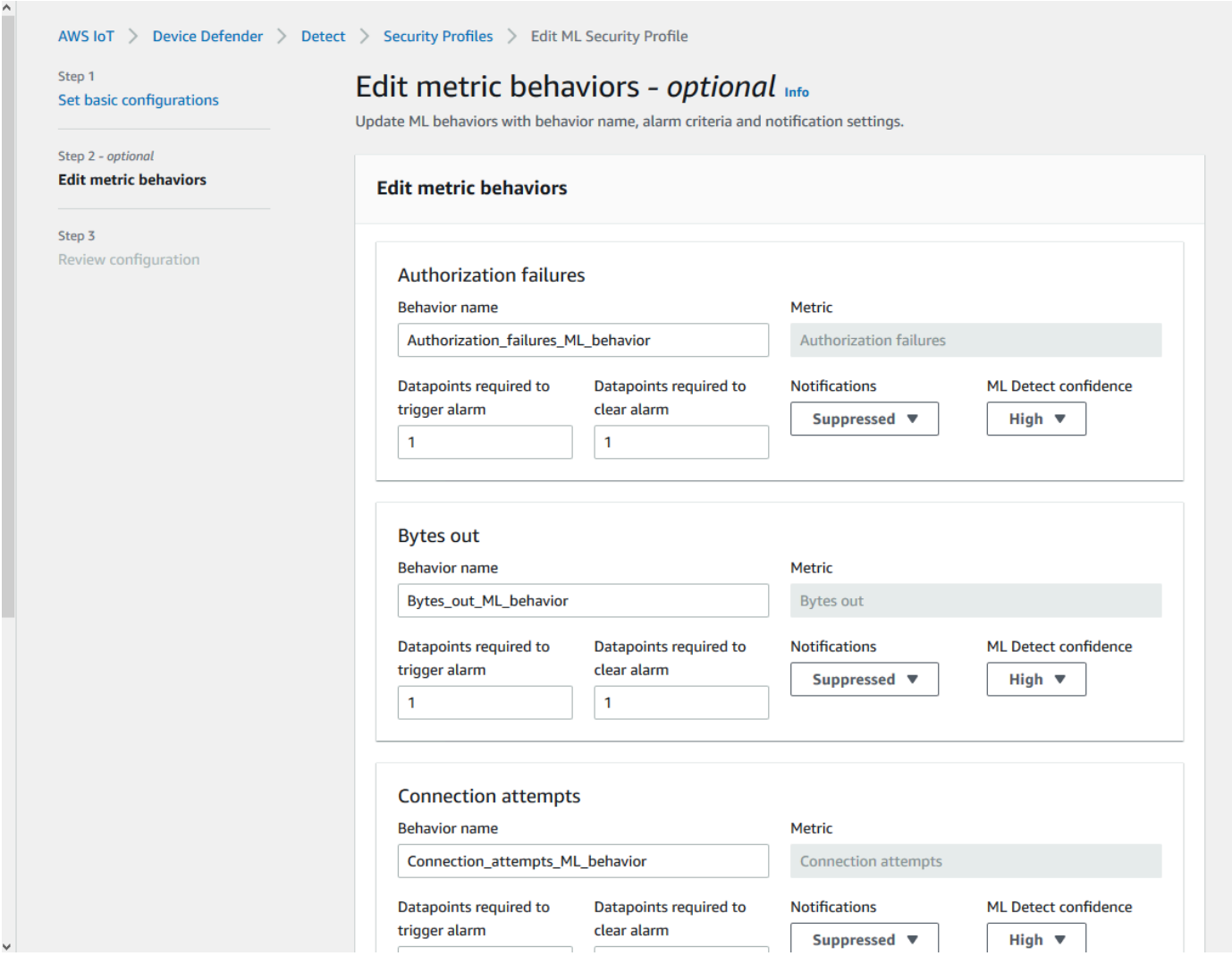


3. 기본 구성 설정에서 보안 프로파일 대상 사물 그룹을 조정하거나 모니터링할 지표를 변경할 수 있습니다.



4. 지표 동작 편집으로 이동하여 다음을 업데이트할 수 있습니다.

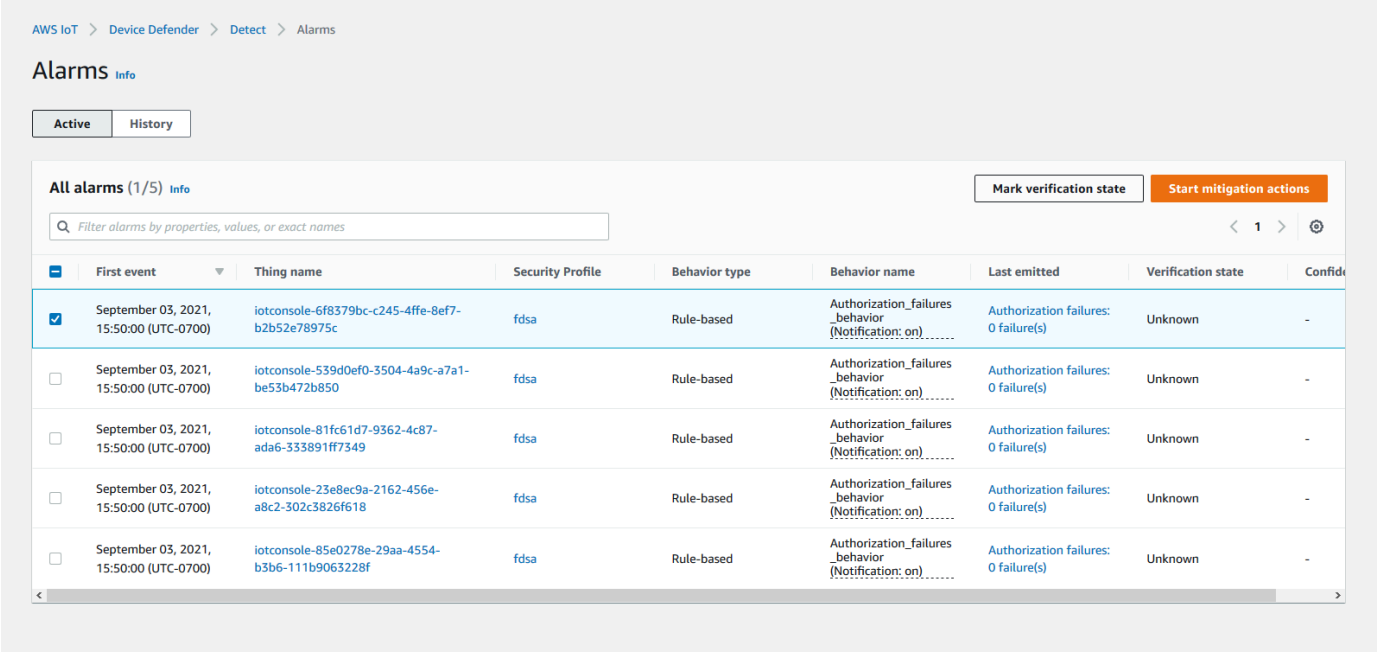
- 경보를 시작하는 데 필요한 ML 모델 데이터 포인트
- 경보를 지우는 데 필요한 ML 모델 데이터 포인트
- ML Detect 신뢰 수준
- ML Detect 알림(예: 금지되지 않음, 금지됨)



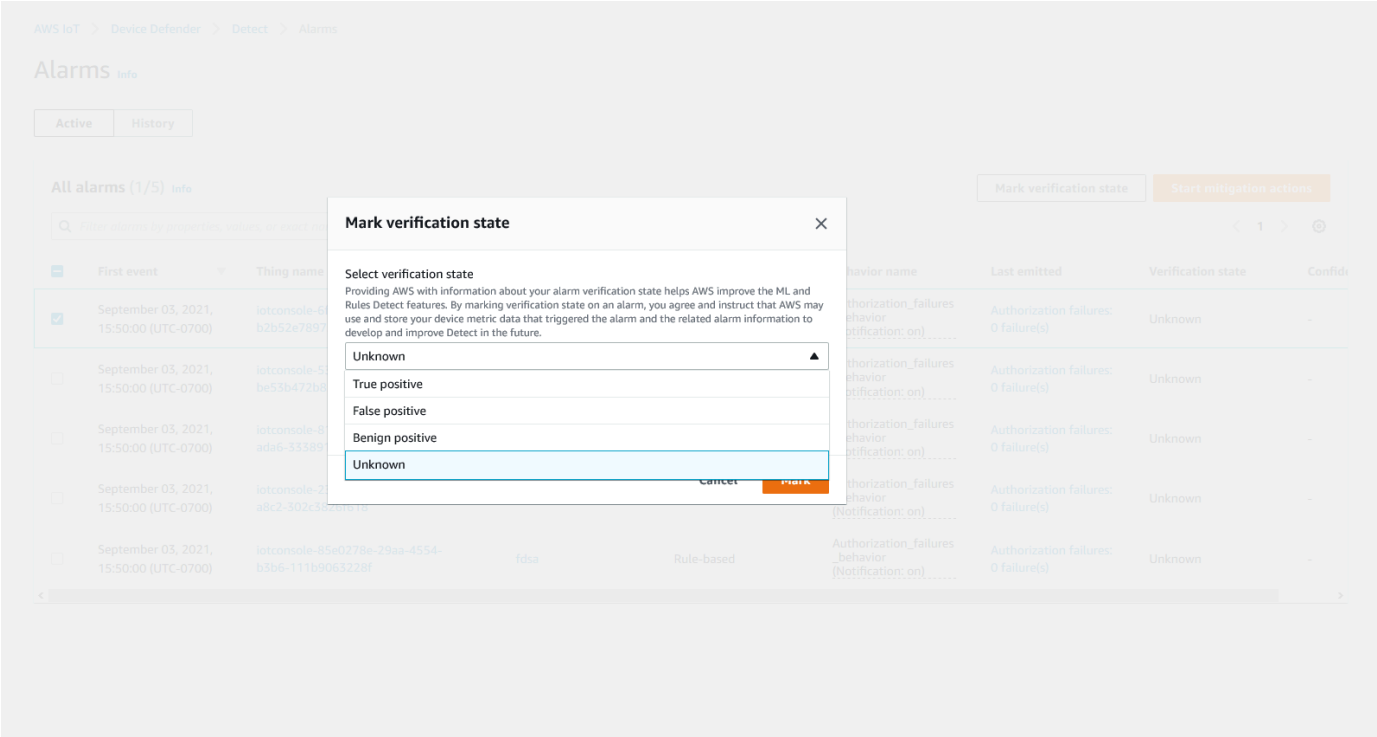
경보의 확인 상태 표시

확인 상태를 설정하고 해당 확인 상태에 대한 설명을 제공하여 경보를 표시합니다. 이를 통해 사용자와 팀이 응답할 필요가 없는 경보를 식별할 수 있습니다.

1. AWS IoT 콘솔의 탐색 창에서 방어(Defend)를 확장한 다음 감지(Detect), 경보(Alarms)를 차례로 선택합니다. 경보를 선택하여 확인 상태를 표시합니다.



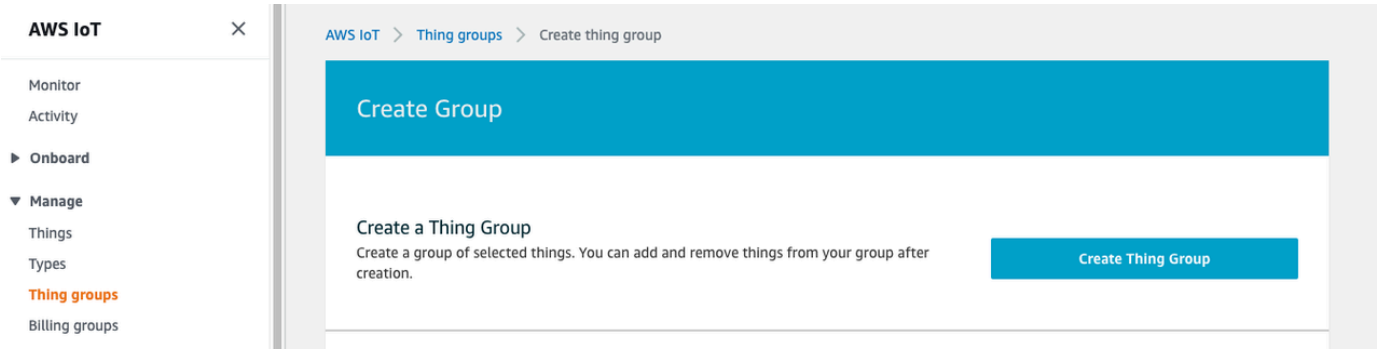
2. 확인 상태 표시(Mark verification state)를 선택합니다. 확인 상태 모달이 열립니다.
3. 적절한 확인 상태를 선택하고 확인 설명(선택 사항)을 입력한 다음 표시(Mark)를 선택합니다. 이 작업은 선택한 경보에 확인 상태 및 설명을 할당합니다.



식별된 디바이스 문제 완화

1. (선택 사항) 격리 완화 작업을 설정하기 전에 위반 중인 디바이스를 이동할 격리 그룹을 설정해 보겠습니다. 기존 그룹을 사용할 수 있습니다.
2. 관리, 사물 그룹, 사물 그룹 생성으로 차례로 이동합니다. 사물 그룹의 이름을 지정합니다. 이 자습서에서는 사물 그룹의 이름을 `Quarantine_group`(으)로 지정합니다. 사물 그룹, 보안에서 사물 그룹에 다음 정책을 적용합니다.

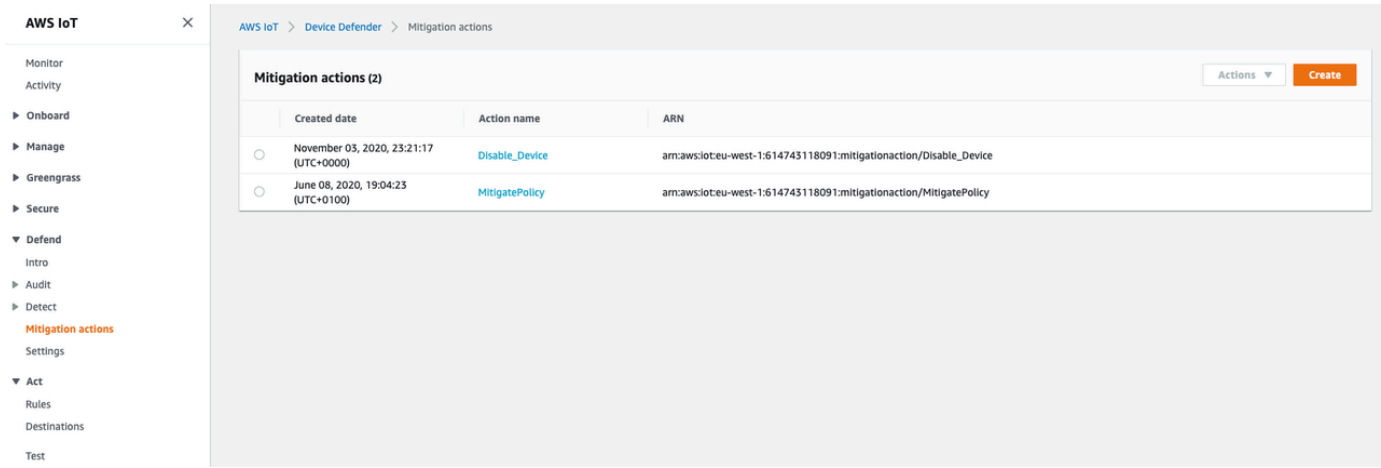
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "iot:*",
      "Resource": "*",
    }
  ]
}
```



완료했으면 사물 그룹 만들기를 선택합니다.

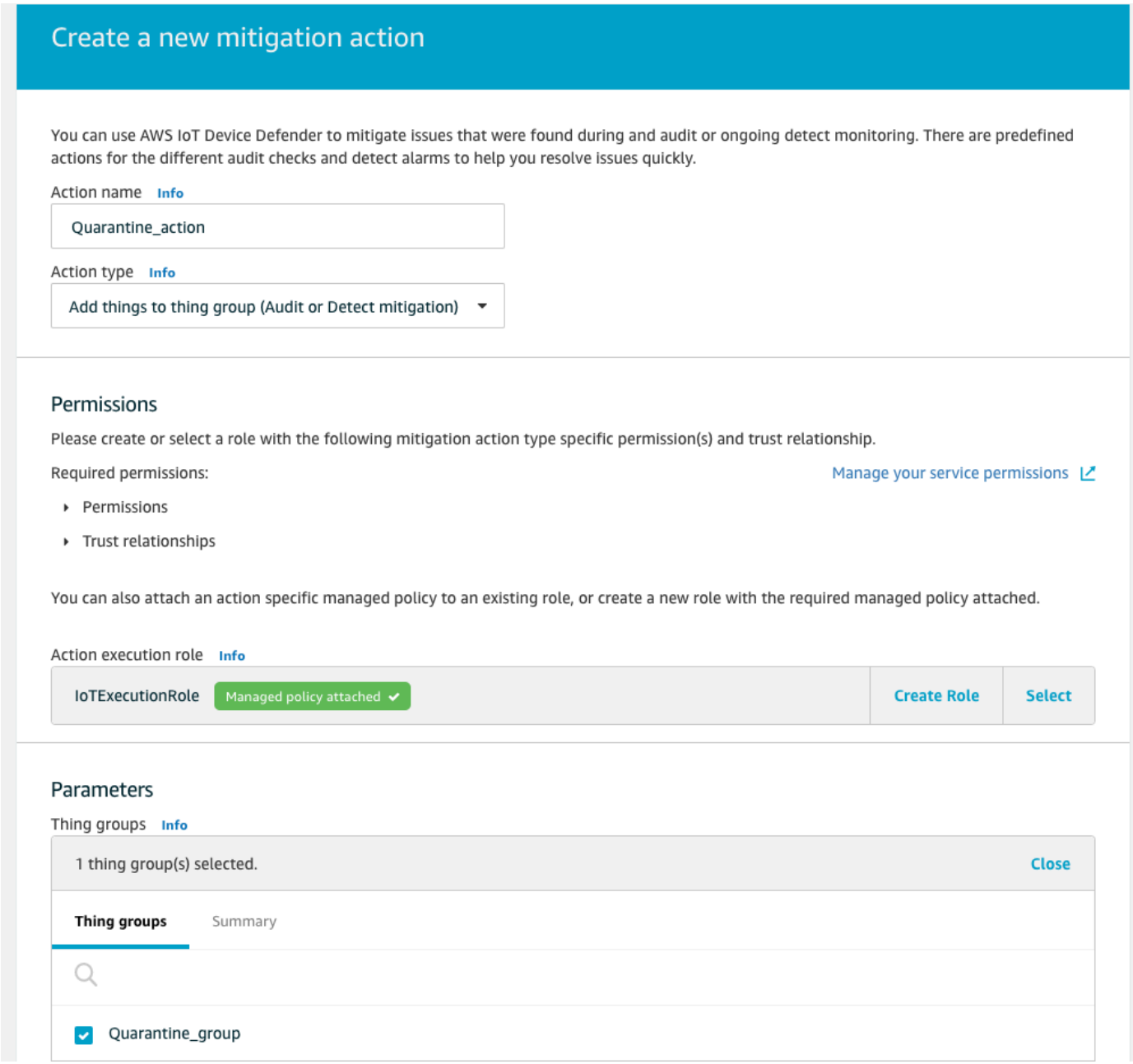
3. 이제 사물 그룹을 생성했으므로 경보가 발생하는 디바이스를 `Quarantine_group`으로 이동하는 완화 작업을 생성해 보겠습니다.

   방어, 완화 작업에서 생성을 선택합니다.



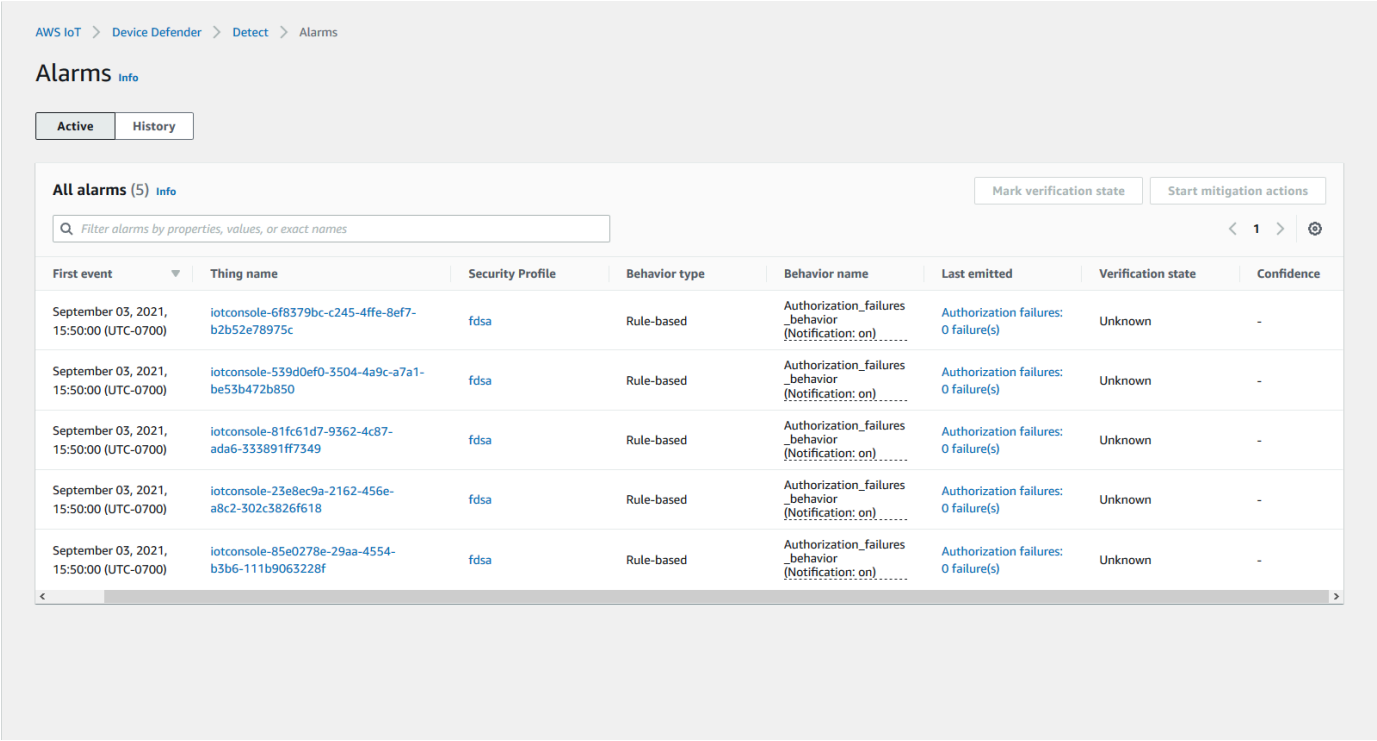
4. 새 완화 작업 생성 페이지에서 다음 정보를 입력합니다.

   - 작업 이름: 완화 작업의 이름을 지정합니다(예:**`Quarantine_action`**).
   - 작업 유형: 작업 유형을 선택합니다. 여기서는 사물 그룹에 사물 추가(감사 또는 감지 완화)를 선택할 것입니다.
   - 작업 실행 역할: 역할을 생성하거나 이전에 생성한 기존 역할을 선택합니다.
   - 파라미터: 사물 그룹을 선택합니다. 이전에 생성한 `Quarantine_group`을(를) 사용할 수 있습니다.



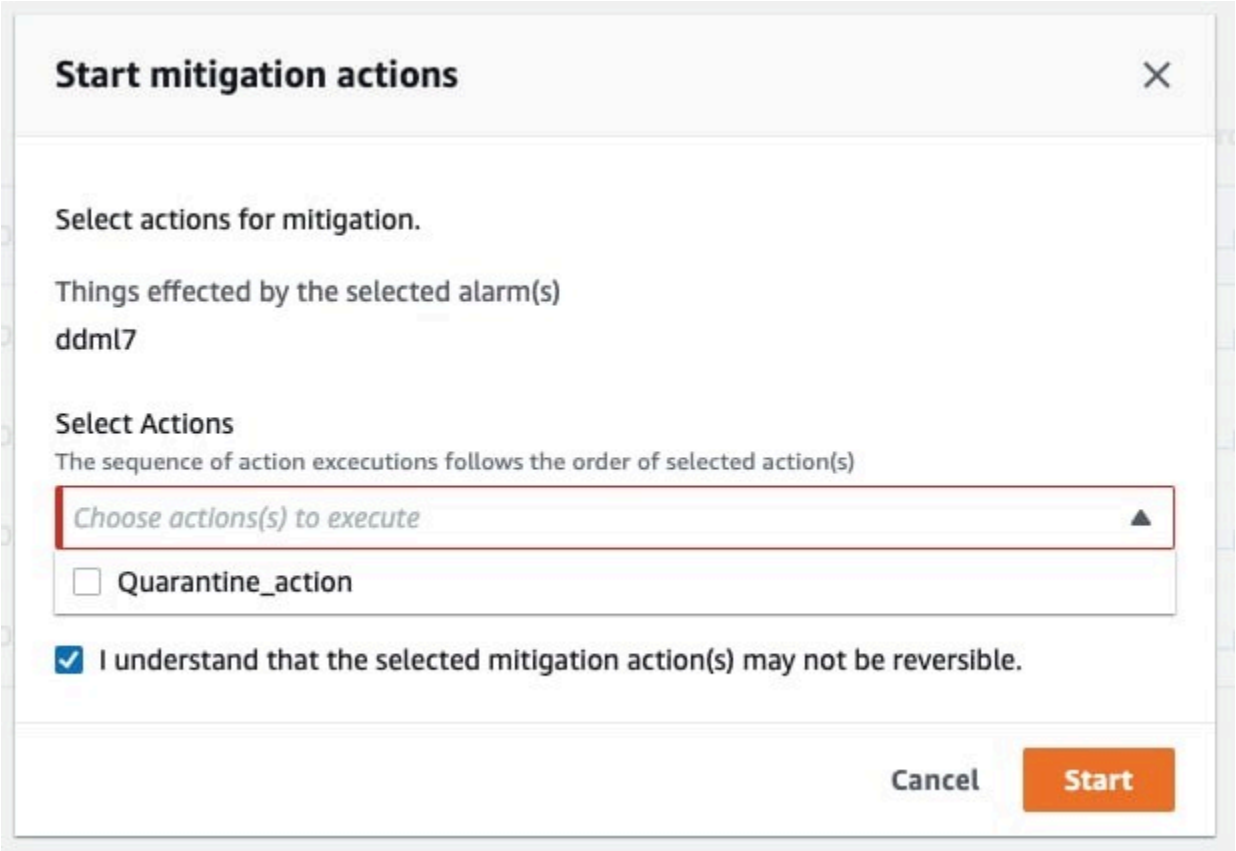
완료되면 저장을 선택합니다. 이제 경보에 있는 디바이스를 격리 사물 그룹으로 이동하는 완화 작업과 조사하는 동안 디바이스를 격리하는 완화 작업이 있습니다.

5. Defender, 감지, 경보로 차례로 이동합니다. 활성 상태 아래에서 어떤 디바이스가 경보 상태에 있는지 확인할 수 있습니다.

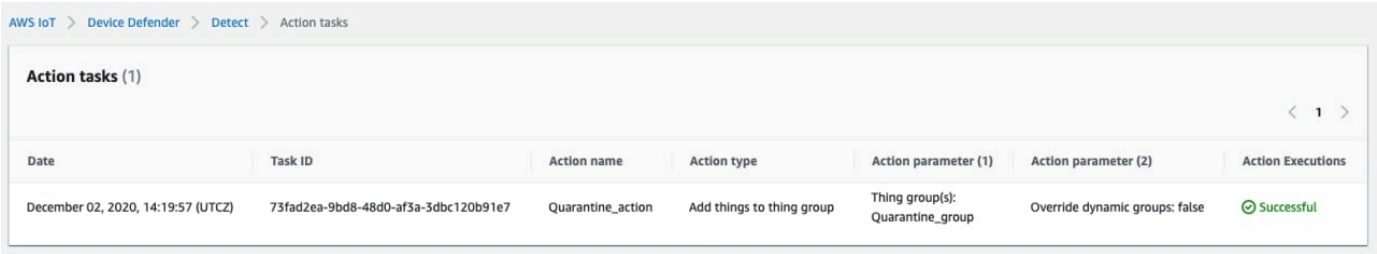


격리 그룹으로 이동할 디바이스를 선택하고 완화 작업 시작을 선택합니다.

6. 완화 작업 시작, 시작 작업에서, 앞에서 생성한 완화 작업을 선택합니다. 예를 들어, **`Quarantine_action`**을 선택한 다음 시작을 선택합니다. 작업 태스크 페이지가 열립니다.



7. 이제 디바이스는 **`Quarantine_group`**에 격리되고 경보를 설정한 문제의 근본 원인을 조사할 수 있습니다. 조사를 완료한 후 디바이스를 사물 그룹 밖으로 이동하거나 추가 작업을 수행할 수 있습니다.



CLI에서 ML 검색을 사용하는 방법

다음은 CLI를 사용하여 ML 검색을 설정하는 방법을 보여줍니다.

자습서

- ML Detect 활성화
- ML 모델 상태 모니터링

- ML Detect 경보 검토
- ML 경보 미세 조정
- 경보의 확인 상태 표시
- 식별된 디바이스 문제 완화

ML Detect 활성화

다음 절차에서는 AWS CLI에서 ML Detect를 활성화하는 방법을 보여 줍니다.

1. 디바이스가 모델의 지속적인 교육 및 새로 고침을 위한 ML Detect 최소 요구 사항에 정의된 대로 필요한 최소 데이터 포인트를 생성하는지 확인하세요. 데이터 수집을 진행하려면 해당 사물이 보안 프로파일에 연결된 사물 그룹에 있어야 합니다.
2. `create-security-profile` 명령을 사용하여 ML Detect 보안 프로파일을 만듭니다. 다음 예제에서는 전송된 메시지 수, 권한 부여 실패 수, 연결 시도 횟수 및 연결 해제 수를 확인하는 *security-profile-for-smart-lights*라는 보안 프로파일을 생성합니다. 이 예에서는 `mlDetectionConfig`을(를) 사용하여 지표가 ML Detect 모델을 사용하도록 설정합니다.

```
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
  }]'
```

출력:

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

3. 그런 다음 보안 프로파일을 하나 또는 여러 개의 사물 그룹과 연결합니다. attach-security-profile 명령을 사용하여 사물 그룹을 보안 프로파일에 연결합니다. 다음 예에서는 *ML_Detect_beta_static_group*이라는 사물 그룹을 *security-profile-for-smart-lights* 보안 프로파일과 연결합니다.

```
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
```

출력:

없음.

4. 전체 보안 프로파일을 만든 후 ML 모델이 교육을 시작합니다. 초기 ML 모델 훈련 및 구축을 완료하는 데 14일이 걸립니다. 14일 후 디바이스에 비정상적인 활동이 있는 경우 경보가 표시될 것으로 예상할 수 있습니다.

ML 모델 상태 모니터링

다음 절차에서는 진행 중인 교육 중인 ML 모델을 모니터링하는 방법을 소개합니다.

- get-behavior-model-training-summaries 명령을 사용하여 ML 모델의 진행 상황을 볼 수 있습니다. 다음 예는 *security-profile-for-smart-lights* 보안 프로파일에 대한 ML 모델 훈련 진행률 요약을 가져옵니다. `modelStatus`는 모델이 교육을 완료했거나 특정 동작에 대한 빌드를 보류 중인지 여부를 보여줍니다.

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

출력:

```
{
    "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**
>
> 모델이 예상대로 진행되지 않는 경우 디바이스가 최소 요구 사항을 충족하는지 확인합니다.

ML Detect 경보 검토

ML 모델을 빌드하고 데이터 평가를 위해 준비한 후에는 모델에서 추론하는 경보를 정기적으로 볼 수 있습니다. 다음 절차에서는 AWS CLI에서 경보를 보는 방법을 보여 줍니다.

- 모든 활성 경보를 보려면 `list-active-violations` 명령을 사용합니다.

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

출력:

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

또는 `list-violation-events` 명령을 사용하여 주어진 기간 동안 발견된 모든 위반을 볼 수 있습니다. 다음 예는 2020년 9월 22일 5:42:13 GMT부터 2020년 10월 26일 5:42:13 GMT까지의 위반 이벤트를 나열합니다.

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

출력:

```
{
    "violationEvents": [
        {
            "violationId": "1448be98c09c3d4ab7cb9b6f3ece65d6",
            "thingName": "lightbulb-1",
            "securityProfileName": "security-profile-for-smart-lights",
            "behavior": {
                "name": "LowConfidence_MladBehavior_MessagesSent",
                "metric": "aws:num-messages-sent",
                "criteria": {
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1,
                    "mlDetectionConfig": {
```

```
                        "confidenceLevel": "HIGH"
                    }
                },
                "suppressAlerts": true
            },
            "violationEventType": "alarm-invalidated",
            "violationEventTime": 1600780245.29
        },
        {
            "violationId": "df4537569ef23efb1c029a433ae84b52",
            "thingName": "lightbulb-2",
            "securityProfileName": "security-profile-for-smart-lights",
            "behavior": {
                "name": "LowConfidence_MladBehavior_MessagesSent",
                "metric": "aws:num-messages-sent",
                "criteria": {
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1,
                    "mlDetectionConfig": {
                        "confidenceLevel": "HIGH"
                    }
                },
                "suppressAlerts": true
            },
            "violationEventType": "alarm-invalidated",
            "violationEventTime": 1600780245.281
        }
    ],
    "nextToken":
 "Amo6XIUrsOohsojuIG6TuwSR3X9iUvH2OCksBZg6bed2j21VSnD1uP1pflxKX1+a3cvBRSosIB0xFv40kM6RYBknZ
vxabMe/ZW31Ps/WiZHlr9Wg7R7eEGli59IJ/U0iBQ1McP/ht0E2XA2TTIvYeMmKQQPsRj/
eoV9j7P/wveu7skNGepU/mvpV0O2Ap7hnV5U+Prx/9+iJA/341va
+pQww7jpUeHmJN9Hw4MqW0ysw0Ry3w38hOQWEpz2xwFWAxAARxeIxCxt5c37RK/lRZBlhYqoB
+w2PZ74730h8pICGY4gktJxkwHyyRabpSM/G/f5DFrD9O5v8idkTZzBxW2jrbzSUIdafPtsZHL/
yAMKr3HAKtaABz2nTsOBNre7X2d/jIjjarhon0Dh9l+8I9Y5Ey
+DIFBcqFTvhibKAafQt3gs6CUiqHdWiCenfJyb8whmDE2qxvdxGElGmRb
+k6kuN5jrZxxw95gzfYDgRHv11iEn8h1qZLD0czkIFBpMppHj9cetHPvM
+qffXGAzKi8tL6eQuCdMLXmVE3jbqcJcjk9ItnaYJi5zKDz9FVbrz9qZZPtZJFHp"
}
```

ML 경보 미세 조정

ML 모델을 빌드하고 데이터 평가를 위한 준비가 완료되면 보안 프로파일의 ML 동작 설정을 업데이트하여 구성을 변경할 수 있습니다. 다음 절차에서는 AWS CLI에서 보안 프로파일의 ML 동작 설정을 업데이트하는 방법을 보여 줍니다.

- 보안 프로파일의 ML 동작 설정을 변경하려면 `update-security-profile` 명령을 사용하세요. 다음 예에서는 몇 가지 동작의 `confidenceLevel`을(를) 변경하여 *`security-profile-for-smart-lights`* 보안 프로파일의 동작을 업데이트하고 모든 동작에 대한 알림을 금지하지 않습니다.

```
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

    }]'
```

출력:

```
 {
    "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
}
```

경보의 확인 상태 표시

경보를 분류하고 이상을 조사하는 데 도움이 되도록 경보에 확인 상태를 표시할 수 있습니다.

- 경보에 확인 상태와 해당 상태에 대한 설명을 표시합니다. 예를 들어 경보의 확인 상태를 거짓 긍정으로 설정하려면 다음 명령을 사용합니다.

```
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
```

출력:

없음.

식별된 디바이스 문제 완화

1. create-thing-group 명령을 사용하여 완화 작업을 위한 사물 그룹을 생성합니다. 다음 예에서는 ThingGroupForDetectMitigationAction이라는 사물 그룹을 생성합니다.

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

출력:

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

2. 다음으로 create-mitigation-action 명령을 사용하여 완화 작업을 생성합니다. 다음 예에서는 완화 작업을 적용하는 데 사용되는 IAM 역할의 ARN을 통해 detect_mitigation_action이라는 완화 작업을 생성합니다. 작업 유형과 해당 작업의 파라미터도 정의합니다. 이 경우, 완화 작업으로 인해 사물은 이전에 생성된 ThingGroupForDetectMitigationAction이라는 사물 그룹으로 이동됩니다.

```
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
     }
 }'
```

출력:

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

3. start-detect-mitigation-actions-task 명령을 사용하여 완화 작업 태스크를 시작합니다. task-id,target 및 actions은(는) 필수 파라미터입니다.

```
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
```

출력:

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

4. (선택 사항) 태스크에 포함된 완화 작업 실행을 보려면 list-detect-mitigation-actions-executions 명령을 사용합니다.

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

출력:

```
{
    "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. (선택 사항) describe-detect-mitigation-actions-task 명령을 사용하여 완화 작업 태스크에 대한 정보를 확인합니다.

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

출력:

```
{
    "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. (선택 사항) 완화 작업 태스크 목록을 가져오려면 list-detect-mitigation-actions-tasks 명령을 사용합니다.

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

출력:

```
{
    "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. (선택 사항) 완화 작업 태스크 목록을 취소하려면 cancel-detect-mitigation-actions-task 명령을 사용합니다.

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

출력:

없음.

AWS IoT Device Defender 감사 결과를 보는 시기와 방법 사용자 지정

AWS IoT Device Defender 감사는 정기적인 보안 검사를 제공하여 AWS IoT 디바이스와 리소스가 모범 사례를 따르고 있는지 확인합니다. 각 검사에 대해 감사 결과는 준수 또는 비준수로 분류되며, 준수하지 않으면 콘솔 경고 아이콘이 표시됩니다. 알려진 문제의 불필요한 반복 표시를 줄이기 위한 감사 결과 금지 기능을 사용하여 일시적으로 이러한 비준수 알림이 표시되지 않도록 설정할 수 있습니다.

미리 결정된 기간 동안 특정 리소스 또는 계정에 대한 특정 감사 검사를 금지(suppress)할 수 있습니다. 금지된 감사 검사 결과는 준수 및 비준수 범주와는 별개로 금지된 결과로 분류됩니다. 이 새 범주는 비준수 결과와 같은 경보를 트리거하지 않습니다. 이를 통해 알려진 유지 관리 기간 동안 또는 업데이트가 완료되도록 예약될 때까지 비준수 알림 방해를 줄일 수 있습니다.

시작하기

다음 단원에서는 감사 결과 금지를 사용하여 콘솔 및 CLI에서 `Device certificate expiring` 검사를 금지하는 방법을 자세히 설명합니다. 데모 중 하나를 따르려는 경우 먼저 Device Defender에서 감지할 수 있는 만료 인증서를 두 개 만들어야 합니다.

다음 단계에 따라 인증서를 만듭니다.

- AWS IoT Core 개발자 안내서의 CA 인증서 생성 및 등록
- CA 인증서를 사용하여 클라이언트 인증서를 생성합니다. 3단계에서 `days` 파라미터를 **1**(으)로 설정합니다.

CLI를 사용하여 인증서를 생성하는 경우 다음 명령을 입력합니다.

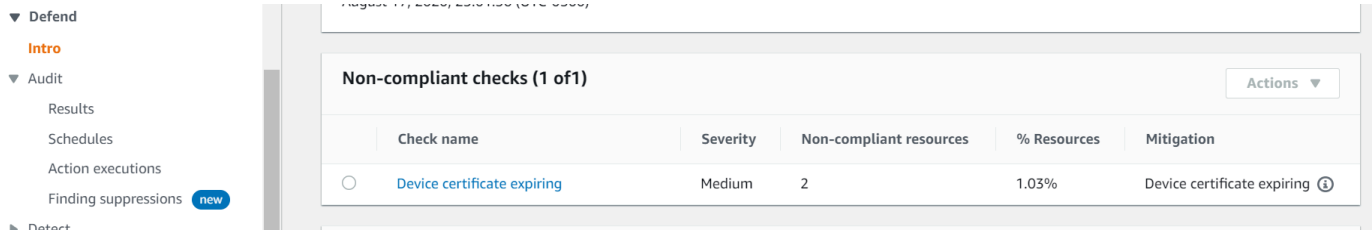
```
openssl x509 -req \
    -in device_cert_csr_filename \
    -CA root_ca_pem_filename \
    -CAkey root_ca_key_filename \
    -CAcreateserial \
    -out device_cert_pem_filename \
    -days 1 -sha256
```

콘솔에서 감사 결과 사용자 지정

다음 연습에서는 비준수 감사 검사를 트리거하는 두 개의 만료된 디바이스 인증서가 있는 계정을 사용합니다. 이 시나리오에서는 개발자가 문제를 해결할 새 기능을 테스트하기 때문에 경고를 비활성화하려고 합니다. 감사 결과가 다음 주에 비준수되는 것을 방지하기 위해 각 인증서에 대해 감사 결과 금지를 생성합니다.

1. 만료된 디바이스 인증서 검사가 비준수임을 보여주기 위해 먼저 온디맨드 감사를 실행합니다.

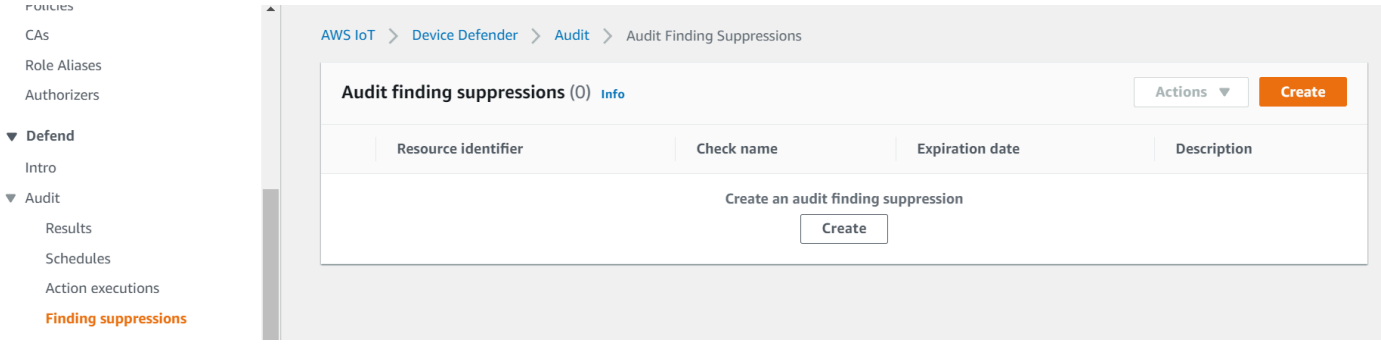
   AWS IoT 콘솔의 왼쪽 사이드바에서 방어를 선택한 다음 감사, 결과를 차례로 선택합니다. 감사 결과 페이지에서 생성을 선택합니다. 새 감사 생성 창이 열립니다. 생성을 선택합니다.



온디맨드 감사 결과에서 “디바이스 인증서 만료”가 두 리소스를 준수하지 않음을 알 수 있습니다.

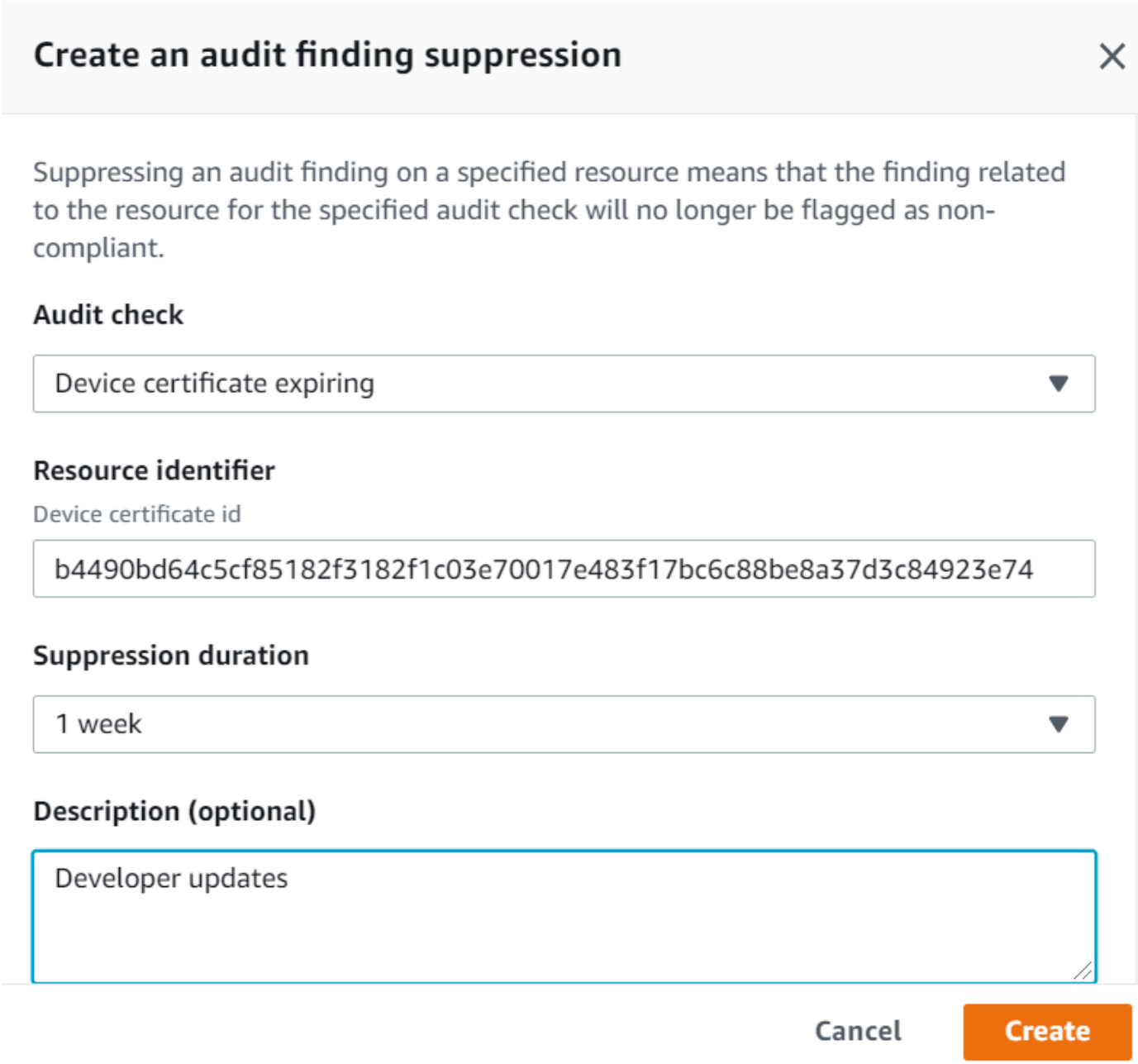
2. 이제 개발자가 경고를 수정할 새로운 기능을 테스트하고 있기 때문에 “디바이스 인증서 만료” 비준수 검사 경고를 비활성화하려고합니다.

   왼쪽 사이드바의 방어에서 감사를 선택한 다음 결과 금지를 선택합니다. 감사 결과 금지 페이지에서 생성을 선택합니다.



3. 감사 결과 금지 생성 창에서 다음을 작성해야합니다.

   - 감사 검사: 금지하려는 감사 검사이므로 `Device certificate expiring`을(를) 선택합니다.
   - 리소스 식별자: 감사 결과를 금지하려는 인증서 중 하나의 디바이스 인증서 ID를 입력합니다.
   - 금지 기간: `Device certificate expiring` 감사 검사를 금지하고자 하는 기간이므로 `1 week`을(를) 선택합니다.
   - 설명(선택 사항): 이 감사 결과를 금지하는 이유를 설명하는 메모를 추가합니다.

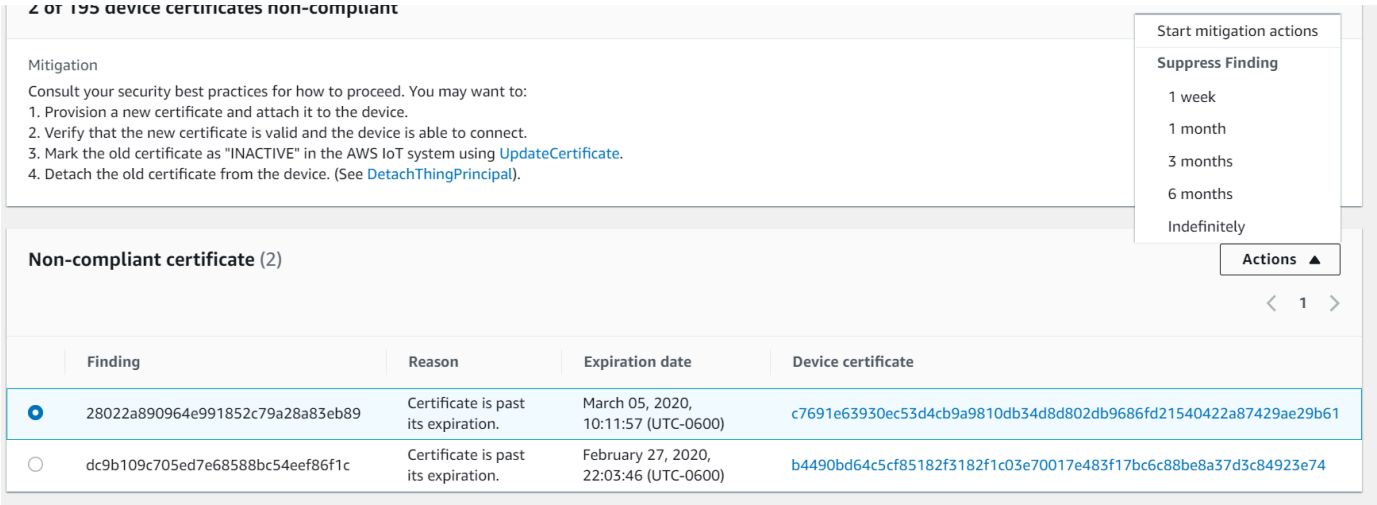


필드를 입력한 후 생성을 선택합니다. 감사 결과 금지가 생성된 후 성공 배너가 표시됩니다.

4. 인증서 중 하나에 대한 감사 결과를 금지했으므로 이제 두 번째 인증서에 대한 감사 결과를 금지해야 합니다. 3단계에서 사용한 것과 동일한 금지 방법을 사용할 수 있지만 데모 목적으로 다른 방법을 사용할 것입니다.

왼쪽 사이드바의 방어에서 감사를 선택한 다음 결과를 선택합니다. 감사 결과 페이지에서 비준수 리소스의 감사를 선택합니다. 그런 다음 비준수 검사에서 리소스를 선택합니다. 이 경우 “디바이스 인증서 만료”를 선택합니다.

5. 디바이스 인증서 만료 페이지의 비준수 정책에서 금지해야 하는 결과 옆에 있는 옵션 버튼을 선택합니다. 그런 다음 작업 드롭다운 메뉴를 선택한 다음 결과를 금지하고 싶은 기간을 선택합니다. 이 경우, 다른 인증서와 마찬가지로 1 week을(를) 선택합니다. 금지 확인 창에서 금지 활성화를 선택합니다.



감사 결과 금지가 생성된 후 성공 배너가 표시됩니다. 이제 개발자가 경고를 해결하기 위한 솔루션을 개발하는 동안 두 감사 결과가 모두 1주일 동안 금지되었습니다.

CLI에서 감사 결과 사용자 지정

다음 연습에서는 비준수 감사 검사를 트리거하는 디바이스 인증서가 만료된 계정을 사용합니다. 이 시나리오에서는 개발자가 문제를 해결할 새 기능을 테스트하기 때문에 경고를 비활성화하려고 합니다. 감사 결과가 다음 주에 비준수되는 것을 방지하기 위해 인증서에 대해 감사 결과 금지를 생성합니다.

다음 CLI 명령을 사용합니다.

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

1. 다음 명령을 사용하여 감사를 활성화합니다.

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

출력:

없음.

2. 다음 명령을 사용하여 DEVICE_CERTIFICATE_EXPIRING_CHECK 감사 검사를 대상으로 하는 온디맨드 감사를 실행합니다.

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

출력:

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

3. describe-account-audit-configuration 명령을 사용하여 감사 구성을 설명합니다. DEVICE_CERTIFICATE_EXPIRING_CHECK에 대한 감사 검사를 설정했는지 확인하고 싶습니다.

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

출력:

```
{
    "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_CHECK은(는) true의 값을 가져야 합니다.

4. list-audit-task 명령을 사용하여 완료된 감사 태스크를 식별할 수 있습니다.

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

출력:

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

1단계에서 실행한 감사의 taskId은(는) COMPLETED의 taskStatus여야 합니다.

5. 이전 단계의 taskId 출력을 사용하여 완료된 감사에 대한 세부 정보를 확인하려면 describe-audit-task 명령을 사용합니다. 이 명령은 감사에 대한 세부 정보를 나열합니다.

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

출력:

```
{
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK",
    "taskStartTime": 1596168096.157,
    "taskStatistics": {
        "totalChecks": 1,
        "inProgressChecks": 0,
        "waitingForDataCollectionChecks": 0,
        "compliantChecks": 0,
        "nonCompliantChecks": 1,
```

```
        "failedChecks": 0,
        "canceledChecks": 0
    },
    "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
    "auditDetails": {
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "checkRunStatus": "COMPLETED_NON_COMPLIANT",
            "checkCompliant": false,
            "totalResourcesCount": 195,
            "nonCompliantResourcesCount": 2
        }
    }
}
```

6. list-audit-findings 명령을 사용하여 비준수 인증서 ID를 찾으면 이 리소스에 대한 감사 알림을 일시 중지할 수 있습니다.

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

출력:

```
{
    "findings": [
        {
            "findingId": "296ccd39f806bf9d8f8de20d0ceb33a1",
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "taskStartTime": 1596168096.157,
            "findingTime": 1596168096.651,
            "severity": "MEDIUM",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId": "b4490<shortened>"
                },
                "additionalInfo": {
                "EXPIRATION_TIME": "1582862626000"
                }
            },
            "reasonForNonCompliance": "Certificate is past its expiration.",
            "reasonForNonComplianceCode": "CERTIFICATE_PAST_EXPIRATION",
```

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

7. create-audit-suppression 명령을 사용하여 *2020-08-20*까지 ID가 *c7691e<shortened>*인 디바이스 인증서에 대한 DEVICE_CERTIFICATE_EXPIRING_CHECK 감사 검사에 대해 알림을 표시하지 않습니다.

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

8. list-audit-suppression 명령을 사용하여 감사 금지 설정을 확인하고 금지에 대한 세부 정보를 얻을 수 있습니다.

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

출력:

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

9. update-audit-suppression 명령을 사용하여 감사 결과 금지를 업데이트할 수 있습니다. 아래 예제에서는 `expiration-date`을(를) `08/21/20`(으)로 업데이트합니다.

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

10. delete-audit-suppression 명령을 사용하여 감사 결과 금지를 제거할 수 있습니다.

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

삭제를 확인하려면 list-audit-suppressions 명령을 사용합니다.

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

출력:

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

이 자습서에서는 콘솔 및 CLI에서 `Device certificate expiring` 검사를 금지하는 방법을 보여 주었습니다. 감사 결과 금지에 대한 자세한 내용은 감사 결과 금지 단원을 참조하세요.

감사

AWS IoT Device Defender 감사에서는 계정 및 디바이스와 관련된 설정 및 정책에서 보안 조치가 구현되어 있는지 확인합니다. 감사는 보안 모범 사례 또는 액세스 정책(예: 동일한 자격 증명을 사용하는 복수의 디바이스 또는 한 디바이스가 다수의 다른 디바이스에서 데이터를 읽고 업데이트하도록 허용하는 과도하게 허용적인 정책)에서 벗어나는 사항을 감지하도록 도울 수 있습니다. 감사를 필요에 따라 실행(온디맨드 감사)하거나 정기적으로 실행되도록 예약(예정된 감사)할 수 있습니다.

AWS IoT Device Defender 감사에서는 일반적인 IoT 보안 모범 사례 및 디바이스 취약성에 대해 일련의 사전 정의된 점검을 실행합니다. 사전 정의된 점검의 예로는 복수의 디바이스에서 데이터를 읽고 업데이트할 권한을 부여하는 정책, 자격 증명(X.509 인증서)을 공유하는 디바이스 또는 만료되었거나 취소되었지만 여전히 활성 상태인 인증서가 포함됩니다.

문제 심각도

문제 심각도는 식별된 각 규정 미준수 인스턴스와 관련된 문제 수준 및 권장 해결 시간을 나타냅니다.

심각

이 심각도의 규정 미준수 감사 검사는 긴급한 주의가 필요한 문제를 식별합니다. 심각한 문제가 발생하면 특별한 자격 증명이 없는 악의적 행위자가 내부자도 모르는 사이 허술하게 자산에 접근하거나 제어하는 것이 쉬워질 수 있습니다.

높음

이 심각도의 규정 미준수 감사 검사는 심각한 문제가 해결된 후 긴급한 조사 및 수정 계획이 필요합니다. 심각한 문제와 마찬가지로 높은 심각도 문제로 인해 악의적 행위자가 자산에 접근하거나 제어할 수 있는 경우가 종종 있습니다. 하지만 높은 심각도 문제는 도용하기가 더 어렵습니다. 이러한 경우는 특별한 도구, 내부자 지식 또는 특정 설정을 필요로 할 수도 있습니다.

중간

이 심각도의 규정 미준수 감사 검사는 지속적인 보안 상태 유지 관리에서 주의를 기울여야 하는 문제를 나타냅니다. 중간 심각도 문제의 경우 보안 제어 오작동으로 인한 예기치 않은 중단과 같은 부정적인 작동 영향을 줄 수 있습니다. 이러한 문제의 경우에도 나쁜 의도의 행위자에게 자산에 대한 제한된 액세스 또는 제어를 제공하거나 이러한 행위자의 악의적인 작업 일부를 촉진할 수도 있습니다.

낮음

이 심각도의 규정 미준수 감사 검사는 종종 보안 모범 사례가 간과되거나 무시되었음을 나타냅니다. 이러한 경우 자체적으로 보안에 즉각적인 영향을 주지는 않을 수도 있지만 악의적 행위자가 이러한 상태를 도용할 수 있습니다. 중간 심각도 문제와 마찬가지로 낮은 심각도 문제의 경우 지속적인 보안 상태 유지 관리에서 주의를 기울여야 합니다.

다음 단계

수행할 수 있는 감사 검사 유형을 이해하려면 감사 검사 단원을 참조하세요. 감사에 적용되는 서비스 할당량에 대한 자세한 내용은 Service Quotas 섹션을 참조하세요.

감사 검사

> **Note**
>
> 검사를 활성화하면 데이터 수집이 즉시 시작됩니다. 계정에서 수집할 데이터의 양이 많은 경우 검사를 활성화한 후 얼마 동안 검사 결과를 사용할 수 없을 수 있습니다.

지원되는 감사 검사는 다음과 같습니다.

- 활성 디바이스 인증서에 대해 취소된 중간 CA 점검
- 취소된 CA 인증서가 계속 활성 상태
- 공유된 디바이스 인증서
- 디바이스 인증서 키 품질
- CA 인증서 키 품질
- 인증되지 않은 Cognito 역할이 지나치게 허용됨
- 인증된 Cognito 역할이 지나치게 허용됨
- 지나치게 허용적인 AWS IoT 정책
- 잘못 구성되었을 가능성이 있는 AWS IoT 정책
- 역할 별칭이 지나치게 허용됨
- 역할 별칭으로 사용되지 않는 서비스에 대한 액세스 허용
- CA 인증서 만료
- 충돌하는 MQTT 클라이언트

- 디바이스 인증서 만료
- 디바이스 인증서 수명 확인
- 취소된 디바이스 인증서가 계속 활성 상태
- 로깅 비활성화

활성 디바이스 인증서에 대해 취소된 중간 CA 점검

중간 CA를 취소했음에도 불구하고 여전히 활성 상태인 모든 관련 디바이스 인증서를 식별하기 위해 이 점검을 사용합니다.

이 점검은 CLI 및 API에서 `INTERMEDIATE_CA_REVOKED_FOR_ACTIVE_DEVICE_CERTIFICATES_CHECK`와(과) 같이 나타납니다.

심각도: 심각

세부 사항

이 점검에서 규정 미준수가 발견된 경우 다음 사유 코드가 반환됩니다.

- INTERMEDIATE_CA_REVOKED_BY_ISSUER

이것이 중요한 이유

활성 디바이스 인증서에 대해 취소된 중간 CA 점검은 CA 체인에서 중간 발급 CA가 취소된 활성 디바이스 인증서가 AWS IoT Core에 있는지 확인하여 디바이스 아이덴티티 및 신뢰도를 평가합니다.

취소된 중간 CA는 더 이상 CA 체인의 다른 CA 또는 디바이스 인증서에 서명하는 데 사용되어서는 안 됩니다. 중간 CA가 취소된 후 이 CA 인증서를 사용하여 서명된 인증서가 새로 추가된 디바이스에 있는 경우에는 보안 위협이 됩니다.

수정 방법

CA 인증서가 취소된 후 디바이스 인증서 등록 활동을 검토합니다. 그리고 보안 모범 사례를 따라 상황을 완화시킵니다. 수행 가능한 작업은 다음과 같습니다.

1. 영향을 받는 디바이스에 다른 CA에서 서명한 새 인증서를 제공합니다.
2. 새 인증서가 유효하고 디바이스가 해당 인증서를 사용하여 연결할 수 있는지 확인합니다.

3. UpdateCertificate을 사용하여 이전 인증서를 AWS IoT에서 취소됨(REVOKED)으로 표시합니다. 완화 작업을 사용하면 다음을 수행할 수 있습니다.
   - 이 변경사항을 실행하려면 감사 결과에서 `UPDATE_DEVICE_CERTIFICATE` 완화 작업을 적용합니다.
   - 조치를 취할 수 있는 그룹에 디바이스를 추가하려면 `ADD_THINGS_TO_THING_GROUP` 완화 조치를 적용합니다.
   - Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.
   - 중간 CA 인증서가 취소된 후 일정 기간에 대해 디바이스 인증서 등록 활동을 검토하여 해당 기간 동안 이 인증서를 사용하여 발급되었을 수 있는 디바이스 인증서를 취소할 것을 고려합니다. ListRelatedResourcesForAuditFinding을 사용하여 CA 인증서로 서명된 디바이스 인증서를 나열하고 UpdateCertificate를 사용하여 디바이스 인증서를 취소합니다.
   - 기존 인증서를 디바이스에서 분리합니다. (DetachThingPrincipal 참조)

   자세한 내용은 완화 작업 단원을 참조하십시오.

취소된 CA 인증서가 계속 활성 상태

CA 인증서가 취소되었지만 AWS IoT에서 계속 활성 상태입니다.

이 점검은 CLI 및 API에서 `REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK`와(과) 같이 나타납니다.

심각도: 심각

세부 사항

CA 인증서가 발급 기관이 관리하는 인증서 취소 목록에 취소됨으로 표시되지만 AWS IoT에는 여전히 ACTIVE 또는 PENDING_TRANSFER로 표시됩니다.

이 점검에서 규정 미준수 CA 인증서가 발견된 경우 다음 사유 코드가 반환됩니다.

- CERTIFICATE_REVOKED_BY_ISSUER

이것이 중요한 이유

취소된 CA 인증서는 더 이상 디바이스 인증서에 서명하는 데 사용해서는 안 됩니다. 손상되었기 때문에 해당 인증서가 취소되었을 수 있습니다. 새로 추가된 디바이스에 이 CA 인증서를 사용하여 서명된 인증서가 있는 경우에는 보안 위협이 발생할 수 있습니다.

수정 방법

1. UpdateCACertificate를 사용하여 AWS IoT에서 CA 인증서를 비활성(INACTIVE) 상태로 표시합니다. 완화 작업을 사용하면 다음을 수행할 수 있습니다.
   - 이 변경사항을 실행하려면 감사 결과에서 `UPDATE_CA_CERTIFICATE` 완화 작업을 적용합니다.
   - Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

   자세한 내용은 완화 작업 단원을 참조하세요.
2. CA 인증서가 취소된 후 일정 기간에 대해 디바이스 인증서 등록 활동을 검토하여 해당 기간 동안 이 인증서를 사용하여 발급되었을 수 있는 디바이스 인증서를 취소할 것을 고려합니다. ListCertificatesByCA를 사용하여 CA 인증서로 서명된 디바이스 인증서를 나열하고 UpdateCertificate를 사용하여 디바이스 인증서를 취소합니다.

공유된 디바이스 인증서

다중 동시 연결은 동일한 X.509 인증서를 사용하여 AWS IoT에서 인증됩니다.

이 점검은 CLI 및 API에서 `DEVICE_CERTIFICATE_SHARED_CHECK`와(과) 같이 나타납니다.

심각도: 심각

세부 사항

온디맨드 감사의 일부로 이 점검을 수행하면 해당 점검에서는 점검이 실행되기 최대 2시간 전까지 감사 시작 전 31일 동안 디바이스에서 연결하기 위해 사용한 인증서 및 클라이언트 ID를 확인합니다. 예정된 감사의 경우 이 검사는 감사의 마지막 실행 2시간 전부터 감사의 이 인스턴스 시작 2시간 전까지의 데이터를 확인합니다. 점검된 시간 동안 이 상태를 완화하는 단계를 수행한 경우 문제가 지속되는지 여부를 확인하기 위해 동시 연결이 발생한 시기를 기록해 둡니다.

이 점검에서 규정 미준수 인증서가 발견된 경우 다음 사유 코드가 반환됩니다.

- CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES

또한, 이 점검에서 반환되는 결과에는 공유 인증서의 ID, 인증서를 사용하여 연결한 클라이언트의 ID 및 연결/연결 해제 횟수가 포함됩니다. 가장 최근 결과가 먼저 나열됩니다.

이것이 중요한 이유

각 디바이스에는 AWS IoT 인증을 위한 고유한 인증서가 있어야 합니다. 여러 디바이스에서 동일한 인증서가 사용되는 경우 이는 디바이스 하나가 손상되었음을 나타내는 것일 수 있습니다. 해당 자격 증명이 복제되어 향후 시스템을 손상시킬 수 있습니다.

수정 방법

디바이스 인증서가 손상되지 않았는지 확인합니다. 손상되었다면 보안 모범 사례를 따라 상황을 완화시킵니다.

여러 디바이스에 동일한 인증서를 사용하는 경우 다음을 수행할 수 있습니다.

1. 고유한 새 인증서를 프로비저닝하고 각 디바이스에 연결합니다.
2. 새 인증서가 유효하고 디바이스가 해당 인증서를 사용하여 연결할 수 있는지 확인합니다.
3. UpdateCertificate을 사용하여 이전 인증서를 AWS IoT에서 취소됨(REVOKED)으로 표시합니다. 완화 작업을 사용하여 다음을 수행할 수도 있습니다.
   - 이 변경사항을 실행하려면 감사 결과에서 `UPDATE_DEVICE_CERTIFICATE` 완화 작업을 적용합니다.
   - 조치를 취할 수 있는 그룹에 디바이스를 추가하려면 `ADD_THINGS_TO_THING_GROUP` 완화 조치를 적용합니다.
   - Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

   자세한 내용은 완화 작업 단원을 참조하세요.
4. 각 디바이스에서 이전 인증서를 분리합니다.

디바이스 인증서 키 품질

AWS IoT 고객은 종종 AWS IoT 메시지 브로커를 인증하기 위해 X.509 인증서를 사용하는 TLS 상호 인증에 의존합니다. 이러한 인증서 및 해당 인증 기관 인증서를 사용하려면 먼저 해당 AWS IoT 계정에 등록해야 합니다. AWS IoT는 등록 시 이러한 인증서에 대해 기본 안전성 점검을 수행합니다. 이러한 점검에는 다음을 확인합니다.

- 인증서가 유효한 형식이어야 합니다.

- 인증서가 등록된 인증 기관에 의해 서명되어야 합니다.
- 인증서 유효 기간이 남아 있어야 합니다. 즉, 만료되지 않은 상태여야 합니다.
- 인증서의 암호화 키 크기가 최소 필수 크기를 충족해야 합니다. RSA 키의 경우 2048비트 이상이어야 합니다.

이 감사 검사에서는 암호화 키의 품질을 다음과 같이 추가로 테스트해야 합니다.

- CVE-2008-0166 – Debian 기반 운영 체제에서 0.9.8g-9 이전 버전까지의 OpenSSL 0.9.8c-1을 사용하여 키가 생성되었는지 확인합니다. 이러한 OpenSSL 버전은 예측 가능한 숫자를 생성하는 난수 생성기를 사용하므로 원격 공격자가 암호화 키에 대한 무차별 대입 공격을 더 쉽게 수행할 수 있습니다.
- CVE-2017-15361 – TPM(Infineon Trusted Platform Module) 펌웨어(예: 0000000000000422 - 4.34 이전 버전, 000000000000062b - 6.43 이전 버전 및 0000000000008521 - 133.33 이전 버전)에서 Infineon RSA 라이브러리 1.02.013에 의해 키가 생성되었는지 확인합니다. 이 라이브러리는 RSA 키 생성을 잘못 처리하기 때문에 공격자가 표적 공격을 통해 일부 암호화 보호 메커니즘을 더 쉽게 파괴할 수 있습니다. 영향을 받는 기술의 예로는 TPM 1.2의 BitLocker, YubiKey 4(4.3.5 이전) PGP 키 생성 및 Chrome OS의 캐시된 사용자 데이터 암호화 기능이 있습니다.

AWS IoT Device Defender는 이러한 테스트에 실패한 인증서를 규정 미준수 인증서로 보고합니다.

이 점검은 CLI 및 API에서 `DEVICE_CERTIFICATE_KEY_QUALITY_CHECK`와(과) 같이 나타납니다.

심각도: 심각

세부 사항

이 점검은 ACTIVE 또는 PENDING_TRANSFER 상태인 디바이스 인증서에 적용됩니다.

이 점검에서 규정 미준수 인증서가 발견된 경우 다음 사유 코드가 반환됩니다.

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

이것이 중요한 이유

디바이스에서 취약한 인증서를 사용하는 경우 공격자가 해당 디바이스를 더 쉽게 손상시킬 수 있습니다.

수정 방법

디바이스 인증서를 업데이트하여 이러한 취약성을 알려진 취약성으로 바꿉니다.

여러 디바이스에 동일한 인증서를 사용하는 경우 다음과 같이 하는 것이 좋습니다.

1. 고유한 새 인증서를 프로비저닝하고 각 디바이스에 연결합니다.
2. 새 인증서가 유효하고 디바이스가 해당 인증서를 사용하여 연결할 수 있는지 확인합니다.
3. UpdateCertificate을 사용하여 이전 인증서를 AWS IoT에서 취소됨(REVOKED)으로 표시합니다. 완화 작업을 사용하면 다음을 수행할 수 있습니다.
   - 이 변경사항을 실행하려면 감사 결과에서 `UPDATE_DEVICE_CERTIFICATE` 완화 작업을 적용합니다.
   - 조치를 취할 수 있는 그룹에 디바이스를 추가하려면 `ADD_THINGS_TO_THING_GROUP` 완화 조치를 적용합니다.
   - Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

   자세한 내용은 완화 작업 단원을 참조하세요.
4. 각 디바이스에서 이전 인증서를 분리합니다.

CA 인증서 키 품질

AWS IoT 고객은 종종 AWS IoT 메시지 브로커를 인증하기 위해 X.509 인증서를 사용하는 TLS 상호 인증에 의존합니다. 이러한 인증서 및 해당 인증 기관 인증서를 사용하려면 먼저 해당 AWS IoT 계정에 등록해야 합니다. AWS IoT는 등록 시 이러한 인증서에 대해 다음과 같은 기본 안전성 점검을 수행합니다.

- 인증서가 유효한 형식입니다.
- 인증서 유효 기간이 남아 있습니다. 즉, 만료되지 않은 상태입니다.
- 인증서의 암호화 키 크기가 최소 필수 크기를 충족합니다. RSA 키의 경우 2048비트 이상이어야 합니다.

이 감사 검사에서는 암호화 키의 품질을 다음과 같이 추가로 테스트해야 합니다.

- CVE-2008-0166 – Debian 기반 운영 체제에서 0.9.8g-9 이전 버전까지의 OpenSSL 0.9.8c-1을 사용하여 키가 생성되었는지 확인합니다. 이러한 OpenSSL 버전은 예측 가능한 숫자를 생성하는 난수 생

성기를 사용하므로 원격 공격자가 암호화 키에 대한 무차별 대입 공격을 더 쉽게 수행할 수 있습니다.

- CVE-2017-15361 – TPM(Infineon Trusted Platform Module) 펌웨어(예: 0000000000000422 - 4.34 이전 버전, 000000000000062b - 6.43 이전 버전 및 0000000000008521 - 133.33 이전 버전)에서 Infineon RSA 라이브러리 1.02.013에 의해 키가 생성되었는지 확인합니다. 이 라이브러리는 RSA 키 생성을 잘못 처리하기 때문에 공격자가 표적 공격을 통해 일부 암호화 보호 메커니즘을 더 쉽게 파괴할 수 있습니다. 영향을 받는 기술의 예로는 TPM 1.2의 BitLocker, YubiKey 4(4.3.5 이전) PGP 키 생성 및 Chrome OS의 캐시된 사용자 데이터 암호화 기능이 있습니다.

AWS IoT Device Defender는 이러한 테스트에 실패한 인증서를 규정 미준수 인증서로 보고합니다.

이 점검은 CLI 및 API에서 `CA_CERTIFICATE_KEY_QUALITY_CHECK`와(과) 같이 나타납니다.

심각도: 심각

세부 사항

이 점검은 ACTIVE 또는 PENDING_TRANSFER 상태인 CA 인증서에 적용됩니다.

이 점검에서 규정 미준수 인증서가 발견된 경우 다음 사유 코드가 반환됩니다.

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

이것이 중요한 이유

이 CA 인증서를 사용하여 서명된 디바이스가 새로 추가된 경우 보안 위협이 발생할 수 있습니다.

수정 방법

1. UpdateCACertificate를 사용하여 AWS IoT에서 CA 인증서를 비활성(INACTIVE) 상태로 표시합니다. 완화 작업을 사용하면 다음을 수행할 수 있습니다.
   - 이 변경사항을 실행하려면 감사 결과에서 `UPDATE_CA_CERTIFICATE` 완화 작업을 적용합니다.
   - Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

   자세한 내용은 완화 작업 단원을 참조하세요.
2. CA 인증서가 취소된 후 일정 기간에 대해 디바이스 인증서 등록 활동을 검토하여 해당 기간 동안 이 인증서를 사용하여 발급되었을 수 있는 디바이스 인증서를 취소할 것을 고려합

니다. (ListCertificatesByCA를 사용하여 CA 인증서로 서명된 디바이스 인증서를 나열하고 UpdateCertificate를 사용하여 디바이스 인증서를 취소합니다.)

인증되지 않은 Cognito 역할이 지나치게 허용됨

인증되지 않은 Amazon Cognito 자격 증명 풀 역할에 연결된 정책은 너무 허용적인 것으로 간주됩니다. 이는 다음 AWS IoT 작업 중 하나를 수행할 수 있는 권한을 부여하기 때문입니다.

- 사물 관리 또는 수정
- 사물 관리 데이터 읽기
- 비사물 관련 데이터 또는 리소스 관리

또는 광범위한 디바이스에서 다음 AWS IoT 작업을 수행할 수 있는 권한을 부여하기 때문입니다.

- MQTT를 사용하여 예약된 주제(섀도우 또는 작업 실행 데이터 포함) 연결, 게시 또는 구독
- API 명령을 사용하여 섀도우 또는 작업 실행 데이터 읽기 또는 수정

일반적으로 인증되지 않은 Amazon Cognito 자격 증명 풀 역할을 사용하여 연결된 디바이스는 사물별 MQTT 주제를 게시하고 구독하거나, API 명령을 사용하여 섀도우 또는 작업 실행 데이터와 관련된 사물별 데이터를 읽고 수정하는 제한된 권한만 보유해야 합니다.

이 점검은 CLI 및 API에서 `UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`와(과) 같이 나타납니다.

심각도: 심각

세부 사항

이 점검의 경우 AWS IoT Device Defender이(가) 감사 실행 전 31일 동안 AWS IoT 메시지 브로커에 연결하는 데 사용된 모든 Amazon Cognito 자격 증명 풀을 감사합니다. 인증된 또는 인증되지 않은 Amazon Cognito 자격 증명이 연결된 모든 Amazon Cognito 자격 증명 풀이 감사에 포함됩니다.

이 점검에서 규정 미준수 인증되지 않은 Amazon Cognito 자격 증명 풀 역할이 발견된 경우 다음 사유 코드가 반환됩니다.

- ALLOWS_ACCESS_TO_IOT_ADMIN_ACTIONS
- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

이것이 중요한 이유

인증되지 않은 자격 증명은 사용자가 인증하지 않으므로 인증된 Amazon Cognito 자격 증명보다 더 큰 위험을 발생합니다. 인증되지 않은 자격 증명이 손상된 경우 관리 작업을 사용하여 계정 설정을 수정하거나, 리소스를 삭제하거나, 중요한 데이터에 접근할 수 있습니다. 또는 디바이스 설정에 광범위하게 액세스하여 계정의 모든 디바이스에 대한 섀도우 및 작업을 액세스하거나 수정할 수 있습니다. 게스트 사용자는 권한을 사용하여 전체 플릿을 손상시키거나 메시지로 DDOS 공격을 시작할 수 있습니다.

수정 방법

인증되지 않은 Amazon Cognito 자격 증명 풀 역할에 연결된 정책은 디바이스에서 작업을 수행하는 데 필요한 권한만 부여해야 합니다. 다음 단계를 수행하는 것이 좋습니다.

1. 새 규정 준수 역할을 생성합니다.
2. Amazon Cognito 자격 증명 풀을 생성하고 이 풀에 규정 준수 역할을 연결합니다.
3. 자격 증명이 새 풀을 통해 AWS IoT에 액세스할 수 있는지 확인합니다.
4. 확인이 완료되면 규정 준수 역할을 규정 미준수로 플래그가 지정된 Amazon Cognito 자격 증명 풀에 연결합니다.

완화 작업을 사용하면 다음을 수행할 수 있습니다.

- Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하세요.

사물 관리 또는 수정

다음 AWS IoT API 작업은 사물을 관리하거나 수정하는 데 사용됩니다. 인증되지 않은 Amazon Cognito 자격 증명 풀을 통해 연결하는 디바이스에는 이 작업을 수행할 권한을 부여해서는 안 됩니다.

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`
- `DeleteThing`
- `DetachThingPrincipal`

- `ListThings`
- `ListThingsInThingGroup`
- `RegisterThing`
- `RemoveThingFromThingGroup`
- `UpdateThing`
- `UpdateThingGroupsForThing`

단일 리소스에 대해서도 이러한 작업을 수행하는 권한을 부여하는 모든 역할은 규정 미준수로 간주됩니다.

사물 관리 데이터 읽기

다음 AWS IoT API 작업은 사물 데이터를 읽거나 수정하는 데 사용됩니다. 인증되지 않은 Amazon Cognito 자격 증명 풀을 통해 연결되는 디바이스에는 이 작업을 수행할 권한을 부여해서는 안 됩니다.

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

Example

- 규정 미준수:

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

```
    }
  ]
}
```

이렇게 하면 디바이스가 사물 하나에 대한 권한만 부여받은 경우에도 지정된 작업을 수행할 수 있습니다.

비사물 관리

인증되지 않은 Amazon Cognito 자격 증명 풀을 통해 연결되는 디바이스에는 이 섹션에서 설명된 작업 이외의 다른 AWS IoT API 작업을 수행하는 권한을 부여해서는 안 됩니다. 디바이스에서 사용되지 않는 별도의 자격 증명 풀을 생성하여 인증되지 않은 Amazon Cognito 자격 증명 풀을 통해 연결되는 애플리케이션이 있는 계정을 관리할 수 있습니다.

MQTT 주제 구독/게시

MQTT 메시지는 AWS IoT 메시지 브로커를 통해 전송되며, 섀도우 상태 및 작업 실행 상태 액세스 및 수정을 포함하여 다양한 작업을 수행하기 위해 디바이스에서 사용됩니다. MQTT 메시지를 연결, 게시 또는 구독할 디바이스에 권한을 부여하는 정책은 다음과 같이 이러한 작업을 특정 리소스로 제한해야 합니다.

연결

- 규정 미준수:

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

  와일드카드 *를 사용하면 모든 디바이스를 AWS IoT에 연결할 수 있습니다.

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

  조건 키에서 `iot:Connection.Thing.IsAttached`가 true로 설정되지 않는 한 이전 예에서 와일드카드 *와 동일합니다.

- 규정 준수:

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

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

리소스 사양에는 연결하는 데 사용된 디바이스 이름과 일치하는 변수가 포함되어 있습니다. 조건문은 MQTT 클라이언트에서 사용되는 인증서가 이름이 사용된 사물에 연결된 인증서와 일치하는지 확인하여 권한을 추가로 제한합니다.

게시

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 디바이스의 섀도우를 업데이트할 수 있습니다(* = 모든 디바이스).

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

이렇게 하면 디바이스가 모든 디바이스의 섀도우에 대해 읽기, 업데이트 또는 삭제 작업을 수행할 수 있습니다.

- 규정 준수:

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

```
}
```

리소스 사양에 와일드카드가 포함되지만, 연결하는 데 사물 이름이 사용된 디바이스의 섀도우 관련 주제와만 일치합니다.

Subscribe

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 디바이스에 대한 예약된 섀도우 또는 작업 주제를 구독할 수 있습니다.

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

이전 예제와 동일하지만, # 와일드카드를 사용합니다.

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

이렇게 하면 디바이스가 모든 디바이스의 섀도우 업데이트를 확인할 수 있습니다(+ = 모든 디바이스).

- 규정 준수:

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

리소스 사양에 와일드카드가 포함되지만, 연결하는 데 사물 이름이 사용된 디바이스의 섀도우 관련 주제 및 작업 관련 주제와만 일치합니다.

수신

- 규정 준수:

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

디바이스에서 구독할 권한이 있는 주제의 메시지만 수신할 수 있으므로 허용됩니다.

섀도우 또는 작업 데이터 읽기/수정

디바이스에 디바이스 섀도우 또는 작업 실행 데이터에 액세스하거나 이를 수정하는 API 작업을 수행할 권한을 부여하는 정책은 이러한 작업을 특정 리소스로 제한해야 합니다. 다음은 API 작업입니다.

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

Example

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 사물에 대해 지정된 작업을 수행할 수 있습니다.

- 규정 준수:

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

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

이렇게 하면 디바이스가 두 개의 사물에 대해서만 지정된 작업을 수행할 수 있습니다.

인증된 Cognito 역할이 지나치게 허용됨

인증된 Amazon Cognito 자격 증명 풀 역할에 연결된 정책은 과도하게 허용적인 것으로 간주됩니다. 이는 다음 AWS IoT 작업 중 하나를 수행할 수 있는 권한을 부여하기 때문입니다.

- 사물 관리 또는 수정
- 비사물 관련 데이터 또는 리소스 관리

또는 광범위한 디바이스에서 다음 AWS IoT 작업을 수행할 수 있는 권한을 부여하기 때문입니다.

- 사물 관리 데이터 읽기
- MQTT를 사용하여 예약된 주제(섀도우 또는 작업 실행 데이터 포함) 연결/게시/구독
- API 명령을 사용하여 섀도우 또는 작업 실행 데이터 읽기 또는 수정

일반적으로 인증된 Amazon Cognito 자격 증명 풀 역할을 사용하여 연결된 디바이스는 사물별 관리 데이터를 읽거나, 사물별 MQTT 주제를 게시하고 구독하거나, API 명령을 사용하여 섀도우 또는 작업 실행 데이터와 관련된 사물별 데이터를 읽고 수정하는 제한된 권한만 보유해야 합니다.

이 점검은 CLI 및 API에서 `AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`와(과) 같이 나타납니다.

심각도: 심각

세부 사항

이 점검의 경우 AWS IoT Device Defender이(가) 감사 실행 전 31일 동안 AWS IoT 메시지 브로커에 연결하는 데 사용된 모든 Amazon Cognito 자격 증명 풀을 감사합니다. 인증된 또는 인증되지 않은 Amazon Cognito 자격 증명이 연결된 모든 Amazon Cognito 자격 증명 풀이 감사에 포함됩니다.

이 점검에서 규정 미준수 인증된 Amazon Cognito 자격 증명 풀 역할이 발견된 경우 다음 사유 코드가 반환됩니다.

- 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

이것이 중요한 이유

인증된 자격 증명이 손상된 경우 관리 작업을 사용하여 계정 설정을 수정하거나, 리소스를 삭제하거나, 중요한 데이터에 액세스할 수 있습니다.

수정 방법

인증된 Amazon Cognito 자격 증명 풀 역할에 연결된 정책은 디바이스에서 작업을 수행하는 데 필요한 권한만 부여해야 합니다. 다음 단계를 수행하는 것이 좋습니다.

1. 새 규정 준수 역할을 생성합니다.
2. Amazon Cognito 자격 증명 풀을 생성하고 이 풀에 규정 준수 역할을 연결합니다.
3. 자격 증명이 새 풀을 통해 AWS IoT에 액세스할 수 있는지 확인합니다.
4. 확인이 완료되면 역할을 규정 미준수로 플래그가 지정된 Amazon Cognito 자격 증명 풀에 연결합니다.

완화 작업을 사용하면 다음을 수행할 수 있습니다.

- Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하세요.

사물 관리 또는 수정

다음 AWS IoT API 작업은 사물을 관리하거나 수정하는 데 사용되므로 인증된 Amazon Cognito 자격 증명 풀을 통해 연결되는 디바이스에는 이러한 작업을 수행하는 권한을 부여해서는 안 됩니다.

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

단일 리소스에 대해서도 이러한 작업을 수행하는 권한을 부여하는 모든 역할은 규정 미준수로 간주됩니다.

비사물 관리

인증된 Amazon Cognito 자격 증명 풀을 통해 연결되는 디바이스에는 이 섹션에서 설명된 작업 이외의 다른 AWS IoT API 작업을 수행하는 권한을 부여해서는 안 됩니다. 인증된 Amazon Cognito 자격 증명 풀을 통해 연결되는 애플리케이션이 있는 계정을 관리하려면 디바이스에서 사용되지 않는 별도의 자격 증명 풀을 생성합니다.

사물 관리 데이터 읽기

다음 AWS IoT API 작업은 사물 데이터를 읽는 데 사용되므로 인증된 Amazon Cognito 자격 증명 풀을 통해 연결되는 디바이스에는 제한된 사물 집합에서만 이러한 작업을 수행하는 권한을 부여해야 합니다.

- `DescribeThing`
- `ListJobExecutionsForThing`

- `ListThingGroupsForThing`
- `ListThingPrincipals`

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 사물에 대해 지정된 작업을 수행할 수 있습니다.

- 규정 준수:

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

이렇게 하면 디바이스가 한 개의 사물에 대해서만 지정된 작업을 수행할 수 있습니다.

- 규정 준수:

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

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

이는 리소스가 와일드카드(*)로 지정된 경우에도 특정 문자열이 리소스 앞에 나오며, 이름에 지정된 접두사가 있는 사물에 액세스된 사물 집합으로 제한하므로 규정 준수입니다.

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 사물에 대해 지정된 작업을 수행할 수 있습니다.

- 규정 준수:

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

이렇게 하면 디바이스가 한 개의 사물에 대해서만 지정된 작업을 수행할 수 있습니다.

- 규정 준수:

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

이는 리소스가 와일드카드(*)로 지정된 경우에도 특정 문자열이 리소스 앞에 나오며, 이름에 지정된 접두사가 있는 사물에 액세스된 사물 집합으로 제한하므로 규정 준수입니다.

MQTT 주제 구독/게시

MQTT 메시지는 AWS IoT 메시지 브로커를 통해 전송되며, 섀도우 상태와 작업 실행 상태 액세스 및 수정을 비롯하여 다양한 작업을 수행하는 디바이스에서 사용됩니다. MQTT 메시지를 연결, 게시 또는 구독할 디바이스에 권한을 부여하는 정책은 다음과 같이 이러한 작업을 특정 리소스로 제한해야 합니다.

연결

- 규정 미준수:

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

  와일드카드 *를 사용하면 모든 디바이스를 AWS IoT에 연결할 수 있습니다.

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

  조건 키에서 `iot:Connection.Thing.IsAttached`가 true로 설정되지 않는 한 이전 예에서 와일드카드 *와 동일합니다.

- 규정 준수:

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

리소스 사양에는 연결하는 데 사용된 디바이스 이름과 일치하는 변수가 포함되어 있으며, 조건문은 MQTT 클라이언트에서 사용되는 인증서가 이름이 사용된 사물에 연결된 인증서와 일치하는지 확인하여 권한을 추가로 제한합니다.

게시

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 디바이스의 섀도우를 업데이트할 수 있습니다(* = 모든 디바이스).

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

이렇게 하면 디바이스가 모든 디바이스의 섀도우에 대해 읽기/업데이트/삭제를 수행할 수 있습니다.

- 규정 준수:

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

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

리소스 사양에 와일드카드가 포함되지만, 연결하는 데 사물 이름이 사용된 디바이스의 섀도우 관련 주제와만 일치합니다.

Subscribe

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 디바이스에 대한 예약된 섀도우 또는 작업 주제를 구독할 수 있습니다.

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

이전 예제와 동일하지만, # 와일드카드를 사용합니다.

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

이렇게 하면 디바이스가 모든 디바이스의 섀도우 업데이트를 확인할 수 있습니다(+ = 모든 디바이스).

- 규정 준수:

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

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

리소스 사양에 와일드카드가 포함되지만, 연결하는 데 사물 이름이 사용된 디바이스의 섀도우 관련 주제 및 작업 관련 주제와만 일치합니다.

수신

- 규정 준수:

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

디바이스에서 구독할 권한이 있는 주제의 메시지만 받을 수 있으므로 규정 준수입니다.

섀도우 또는 작업 데이터 읽기 또는 수정

디바이스에 디바이스 섀도우 또는 작업 실행 데이터에 액세스하거나 이를 수정하는 API 작업을 수행할 권한을 부여하는 정책은 이러한 작업을 특정 리소스로 제한해야 합니다. 다음은 API 작업입니다.

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

예시

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 사물에 대해 지정된 작업을 수행할 수 있습니다.

- 규정 준수:

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

이렇게 하면 디바이스가 두 개의 사물에 대해서만 지정된 작업을 수행할 수 있습니다.

지나치게 허용적인 AWS IoT 정책

AWS IoT 정책이 너무 광범위하거나 제한되지 않은 권한을 제공합니다. 광범위한 디바이스에 대한 MQTT 메시지를 전송 또는 수신하는 권한을 부여하거나 광범위한 디바이스에 대한 섀도우 및 작업 실행 데이터를 액세스 또는 수정하는 권한을 부여합니다.

일반적으로 디바이스에 대한 정책은 해당 디바이스에만 관련되며 다른 디바이스에는 관련되지 않거나 몇몇 디바이스에만 관련된 리소스에 대한 액세스 권한을 부여해야 합니다. 일부 예외를 제외하고 이러한 정책에서 와일드카드(예: "*")를 사용하여 리소스를 지정하는 것은 너무 광범위하거나 제한되지 않다고 간주됩니다.

이 점검은 CLI 및 API에서 `IOT_POLICY_OVERLY_PERMISSIVE_CHECK`와(과) 같이 나타납니다.

심각도: 심각

세부 사항

이 점검에서 규정 미준수 AWS IoT 정책이 발견된 경우 다음 사유 코드가 반환됩니다.

- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

이것이 중요한 이유

과도하게 허용적인 정책이 있는 인증서, Amazon Cognito 자격 증명 또는 사물 그룹이 손상된 경우 전체 계정의 보안에 영향을 줄 수 있습니다. 해커는 모든 디바이스에 대한 섀도우, 작업 또는 작업 실행을 읽거나 수정할 수 있는 광범위한 액세스 권한을 사용할 수 있습니다. 또는 손상된 인증서를 사용하여 악성 디바이스를 연결하거나 네트워크에 DDOS 공격을 시작할 수도 있습니다.

수정 방법

다음 단계에 따라 사물, 사물 그룹 또는 기타 개체에 연결된 규정 미준수 정책을 수정하세요.

1. CreatePolicyVersion을 사용하여 새 규정 준수 정책 버전을 생성합니다. `setAsDefault` 플래그를 true로 설정합니다. (이로 인해 정책을 사용하는 모든 개체에 이 새 버전이 적용됩니다.)
2. ListTargetsForPolicy를 사용하여 정책이 연결된 대상(인증서, 사물 그룹)의 목록을 가져오고 그룹에 포함되는 디바이스 또는 인증서를 사용하여 연결할 디바이스를 결정합니다.
3. 연결된 모든 디바이스가 AWS IoT에 연결될 수 있는지 확인합니다. 디바이스를 연결할 수 없는 경우 SetPolicyVersion을 사용하여 기본 정책을 이전 버전으로 롤백하고 정책을 수정한 다음 다시 시도하세요.

완화 조치를 사용하면 다음을 수행할 수 있습니다.

- 이 변경사항을 실행하려면 감사 결과에서 `REPLACE_DEFAULT_POLICY_VERSION` 완화 작업을 적용합니다.
- Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하세요.

AWS IoT Core 정책 변수를 사용하여 정책에서 AWS IoT 리소스를 동적으로 참조합니다.

MQTT 권한

MQTT 메시지는 AWS IoT 메시지 브로커를 통해 전송되며, 섀도우 상태 및 작업 실행 상태 액세스 및 수정을 포함하여 다양한 작업을 수행하기 위해 디바이스에서 사용됩니다. MQTT 메시지를 연결, 게시 또는 구독할 디바이스에 권한을 부여하는 정책은 다음과 같이 이러한 작업을 특정 리소스로 제한해야 합니다.

연결

- 규정 미준수:

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

  와일드카드 *를 사용하면 모든 디바이스를 AWS IoT에 연결할 수 있습니다.

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

  조건 키에서 `iot:Connection.Thing.IsAttached`가 true로 설정되지 않는 한 이전 예에서 와일드카드 *와 동일합니다.

- 규정 준수:

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

  리소스 사양에는 연결하는 데 사용된 디바이스 이름과 일치하는 변수가 포함되어 있습니다. 조건 문은 MQTT 클라이언트에서 사용되는 인증서가 이름이 사용된 사물에 연결된 인증서와 일치하는지 확인하여 권한을 추가로 제한합니다.

게시

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 디바이스의 섀도우를 업데이트할 수 있습니다(* = 모든 디바이스).

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

이렇게 하면 디바이스가 모든 디바이스의 섀도우에 대해 읽기, 업데이트 또는 삭제 작업을 수행할 수 있습니다.

- 규정 준수:

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

리소스 사양에 와일드카드가 포함되지만, 연결하는 데 사물 이름이 사용된 디바이스의 섀도우 관련 주제와만 일치합니다.

Subscribe

- 규정 미준수:

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

이렇게 하면 디바이스가 모든 디바이스에 대한 예약된 섀도우 또는 작업 주제를 구독할 수 있습니다.

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

이전 예제와 동일하지만, # 와일드카드를 사용합니다.

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

이렇게 하면 디바이스가 모든 디바이스의 섀도우 업데이트를 확인할 수 있습니다(+ = 모든 디바이스).

- 규정 준수:

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

리소스 사양에 와일드카드가 포함되지만, 연결하는 데 사물 이름이 사용된 디바이스의 섀도우 관련 주제 및 작업 관련 주제와만 일치합니다.

수신

- 규정 준수:

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

디바이스에서 구독할 권한이 있는 주제의 메시지만 받을 수 있으므로 규정 준수입니다.

섀도우 및 작업 권한

디바이스에 디바이스 섀도우 또는 작업 실행 데이터에 액세스하거나 이를 수정하는 API 작업을 수행할 권한을 부여하는 정책은 이러한 작업을 특정 리소스로 제한해야 합니다. 다음은 API 작업입니다.

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

예시

- 규정 미준수:

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

  이렇게 하면 디바이스가 모든 사물에 대해 지정된 작업을 수행할 수 있습니다.

- 규정 준수:

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

```
        "arn:aws:iot:region:account-id:/thing/MyThing2"
      ]
    }
  ]
}
```

이렇게 하면 디바이스가 두 개의 사물에 대해서만 지정된 작업을 수행할 수 있습니다.

잘못 구성되었을 가능성이 있는 AWS IoT 정책

AWS IoT 정책이 잘못 구성되었을 가능성이 있는 것으로 확인되었습니다. 지나치게 허용적인 정책을 비롯하여 정책이 잘못 구성되면 의도하지 않은 리소스에 대한 디바이스 액세스를 허용하는 등의 보안 인시던트가 발생할 수 있습니다.

잘못 구성되었을 가능성이 있는 AWS IoT 정책 점검은 정책을 업데이트하기 전에 의도한 작업만 허용되는지 확인하라는 경고입니다.

이 점검은 CLI 및 API에서 `IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK`와 같이 나타납니다.

심각도: 중간

세부 사항

이 점검에서 잘못 구성되었을 가능성이 있는 AWS IoT 정책이 발견되면 AWS IoT에서 다음 사유 코드가 반환됩니다.

- POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT
- TOPIC_FILTERS_INTENDED_TO_DENY_ALLOWED_USING_WILDCARDS

이것이 중요한 이유

정책을 잘못 구성하면 디바이스에 필요한 것보다 더 많은 권한을 제공하여 의도하지 않은 결과를 초래할 수 있습니다. 리소스에 대한 액세스를 제한하고 보안 위협을 방지하기 위해 정책을 신중하게 고려하는 것이 좋습니다.

정책의 거부 명령문에 MQTT 와일드카드가 포함된 예제

잘못 구성되었을 가능성이 있는 AWS IoT 정책 점검은 거부 명령문에 MQTT 와일드카드 문자(+ 또는 #)가 있는지 검사합니다. 와일드카드는 AWS IoT 정책에서 리터럴 문자열로 취급되므로 정책을 지나치게 허용적으로 만들 수 있습니다.

다음 예시는 정책에서 MQTT 와일드카드 #를 사용하여 `building/control_room` 관련 주제에 대한 구독을 거부하도록 작성되었습니다. 그러나 MQTT 와일드카드는 AWS IoT 정책에서 와일드카드의 의미가 없으며 디바이스가 `building/control_room/data1`을 구독할 수 있습니다.

잘못 구성되었을 가능성이 있는 AWS IoT 정책 점검을 수행하면 이 정책에 사유 코드 `POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT`로 플래그가 지정됩니다.

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

다음은 적절하게 구성된 정책의 예시입니다. 디바이스에는 `building/control_room/`의 하위 주제를 구독할 수 있는 권한이 없으며 `building/control_room/`의 하위 주제에서 메시지를 수신할 수 있는 권한이 없습니다.

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

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

와일드카드를 사용하여 허용을 거부하려고 하는 주제 필터 예시

다음 예시 정책은 `building/control_room/*` 리소스를 거부하여 `building/control_room` 관련 주제에 대한 구독을 거부하도록 작성되었습니다. 그러나 디바이스는 `building/#`에 대한 구독 요청을 보내고 `building/control_room/data1`을 포함하여 `building`과 관련된 모든 주제에서 메시지를 받을 수 있습니다.

잘못 구성되었을 가능성이 있는 AWS IoT 정책 점검을 수행하면 이 정책에 사유 코드 `TOPIC_FILTERS_INTENDED_TO_DENY_ALLOWED_USING_WILDCARDS`로 플래그가 지정됩니다.

다음 예시 정책에는 `building/control_room topics` 주제에 대한 메시지를 수신할 수 있는 권한이 있습니다.

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

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

다음은 적절하게 구성된 정책의 예시입니다. 디바이스에는 building/control_room/의 하위 주제를 구독할 수 있는 권한이 없으며 building/control_room/의 하위 주제에서 메시지를 수신할 수 있는 권한이 없습니다.

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

```
}
```

> **Note**
>
> 이 점검에서는 오탐지(false positive)가 보고될 수 있습니다. 플래그가 지정된 정책을 평가하고 감사 억제를 사용하여 오탐지 리소스를 표시하는 것이 좋습니다.

수정 방법

이 점점은 잘못 구성되었을 가능성이 있는 정책에 플래그를 지정하므로 오탐지가 발생할 수 있습니다. 나중에 플래그가 지정되지 않도록 감사 억제를 사용하여 오탐지를 표시합니다.

또한 다음 단계에 따라 사물, 사물 그룹 또는 기타 개체에 연결된 규정 미준수 정책을 수정할 수 있습니다.

1. CreatePolicyVersion을 사용하여 새 규정 준수 정책 버전을 생성합니다. `setAsDefault` 플래그를 true로 설정합니다. (이로 인해 정책을 사용하는 모든 개체에 이 새 버전이 적용됩니다.)

   일반적인 사용 사례에 대한 AWS IoT 정책 생성 예시는 AWS IoT Core 개발자 안내서의 게시 및 구독 정책 예제를 참조하세요.
2. 연결된 모든 디바이스가 AWS IoT에 연결될 수 있는지 확인합니다. 디바이스를 연결할 수 없는 경우 SetPolicyVersion을 사용하여 기본 정책을 이전 버전으로 롤백하고 정책을 수정한 다음 다시 시도하세요.

완화 조치를 사용하면 다음을 수행할 수 있습니다.

- 이 변경사항을 실행하려면 감사 결과에서 `REPLACE_DEFAULT_POLICY_VERSION` 완화 작업을 적용합니다.
- Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하십시오.

AWS IoT Core 개발자 안내서의 IoT Core 정책 변수를 사용하여 정책에서 AWS IoT 리소스를 동적으로 참조합니다.

역할 별칭이 지나치게 허용됨

AWS IoT 역할 별칭은 연결된 디바이스가 X.509 인증서를 사용하여 AWS IoT에 인증한 다음 AWS IoT 역할 별칭과 연결된 IAM 역할에서 수명이 짧은 AWS 자격 증명을 얻는 메커니즘을 제공합니다. 이러한 자격 증명에 대한 권한의 경우 인증 컨텍스트 변수가 있는 액세스 정책을 사용하여 범위를 축소해야 합니다. 정책이 올바르게 구성되지 않은 경우 권한 공격의 에스컬레이션에 노출될 수 있습니다. 이 감사 검사에서는 AWS IoT 역할 별칭에서 제공하는 임시 자격 증명이 과도하게 허용적이지 않도록 합니다.

다음 조건 중 하나가 검색되면 이 점검이 트리거됩니다.

- 이 정책은 이 역할 별칭(예: "iot:*", "dynamodb:*", "iam:*" 등)에서 작년에 사용한 모든 서비스에 대한 관리 권한을 제공합니다.
- 이 정책은 사물 메타데이터 작업에 대한 광범위한 액세스, 제한된 AWS IoT 작업에 대한 액세스 또는 AWS IoT 데이터 영역 작업에 대한 광범위한 액세스를 제공합니다.
- 이 정책은 "iam", "cloudtrail", "guardduty", "inspector" 또는 "trustedadvisor"와 같은 보안 감사 서비스에 대한 액세스를 제공합니다.

이 점검은 CLI 및 API에서 `IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK`와(과) 같이 나타납니다.

심각도: 심각

세부 사항

이 점검에서 규정 미준수 IoT 정책이 발견되면 다음 사유 코드가 반환됩니다.

- 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

이것이 중요한 이유

권한을 디바이스가 정상적인 작업을 수행하는 데 필요한 권한으로 제한하면 디바이스가 손상된 경우 계정에 대한 위험을 줄일 수 있습니다.

수정 방법

다음 단계에 따라 사물, 사물 그룹 또는 기타 개체에 연결된 규정 미준수 정책을 수정하세요.

1. AWS IoT Core 보안 인증 정보 공급자를 사용하여 AWS 서비스에 대한 직접 호출 승인의 단계에 따라 역할 별칭에 보다 제한적인 정책을 적용하세요.

완화 조치를 사용하면 다음을 수행할 수 있습니다.

- Amazon SNS 메시지에 대한 응답으로 사용자 지정 작업을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하세요.

역할 별칭으로 사용되지 않는 서비스에 대한 액세스 허용

AWS IoT 역할 별칭은 연결된 디바이스가 X.509 인증서를 사용하여 AWS IoT에 인증한 다음 AWS IoT 역할 별칭과 연결된 IAM 역할에서 수명이 짧은 AWS 자격 증명을 얻는 메커니즘을 제공합니다. 이러한 자격 증명에 대한 권한의 경우 인증 컨텍스트 변수가 있는 액세스 정책을 사용하여 범위를 축소해야 합니다. 정책이 올바르게 구성되지 않은 경우 권한 공격의 에스컬레이션에 노출될 수 있습니다. 이 감사 검사에서는 AWS IoT 역할 별칭에서 제공하는 임시 자격 증명이 과도하게 허용적이지 않도록 합니다.

이 점검은 역할 별칭이 작년에 AWS IoT 디바이스에 사용되지 않은 서비스에 액세스할 수 있는 경우 트리거됩니다. 예를 들어, 이 감사에서는 작년에 AWS IoT만 사용한 역할 별칭에 연결된 IAM 역할이 있는데 이 역할에 연결된 정책에서도 `"iam:getRole"` 및 `"dynamodb:PutItem"`에 권한을 부여하는지 보고합니다.

이 점검은 CLI 및 API에서 `IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK`와(과) 같이 나타납니다.

심각도: 중간

세부 사항

이 점검에서 규정 미준수 AWS IoT 정책이 발견되면 다음 사유 코드가 반환됩니다.

- ALLOWS_ACCESS_TO_UNUSED_SERVICES

이것이 중요한 이유

권한을 디바이스가 정상적인 작업을 수행하는 데 필요한 서비스로 제한하면 디바이스가 손상된 경우 계정에 대한 위험을 줄일 수 있습니다.

수정 방법

다음 단계에 따라 사물, 사물 그룹 또는 기타 개체에 연결된 규정 미준수 정책을 수정하세요.

1. AWS IoT Core 보안 인증 정보 공급자를 사용하여 AWS 서비스에 대한 직접 호출 승인의 단계에 따라 역할 별칭에 보다 제한적인 정책을 적용하세요.

완화 조치를 사용하면 다음을 수행할 수 있습니다.

- Amazon SNS 메시지에 대한 응답으로 사용자 지정 작업을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하세요.

CA 인증서 만료

CA 인증서가 30일 이내에 만료되거나 만료되었습니다.

이 점검은 CLI 및 API에서 `CA_CERTIFICATE_EXPIRING_CHECK`와(과) 같이 나타납니다.

심각도: 중간

세부 사항

이 점검은 ACTIVE 또는 PENDING_TRANSFER 상태인 CA 인증서에 적용됩니다.

이 점검에서 규정 미준수 CA 인증서가 발견된 경우 다음 사유 코드가 반환됩니다.

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

이것이 중요한 이유

만료된 CA 인증서는 새 디바이스 인증서에 서명하는 데 사용해서는 안 됩니다.

수정 방법

방법은 보안 모범 사례를 참조하세요. 수행 가능한 작업은 다음과 같습니다.

1. 새 CA 인증서를 AWS IoT에 등록합니다.
2. 새 CA 인증서를 사용하여 디바이스 인증서에 서명할 수 있는지 확인합니다.
3. UpdateCACertificate를 사용하여 AWS IoT에서 이전 CA 인증서를 비활성(INACTIVE)으로 표시합니다. 완화 작업을 사용하여 다음을 수행할 수도 있습니다.
   - 이 변경사항을 실행하려면 감사 결과에서 `UPDATE_CA_CERTIFICATE` 완화 작업을 적용합니다.
   - Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

   자세한 내용은 완화 작업 단원을 참조하세요.

충돌하는 MQTT 클라이언트

여러 디바이스가 동일한 클라이언트 ID를 사용하여 연결됩니다.

이 점검은 CLI 및 API에서 `CONFLICTING_CLIENT_IDS_CHECK`와(과) 같이 나타납니다.

심각도: 높음

세부 사항

동일한 클라이언트 ID를 사용한 다중 연결이 성립되어 이미 연결된 디바이스가 연결 해제되었습니다. MQTT 사양에서는 클라이언트 ID당 활성 연결 하나만 허용하므로, 다른 디바이스가 동일한 클라이언트 ID를 사용하여 연결할 경우 이전 디바이스의 연결이 종료됩니다.

온디맨드 감사의 일부로 이 점검을 수행하면 해당 점검에서는 감사 시작 전 31일 동안 클라이언트 ID가 연결에 사용된 방식을 확인합니다. 예정된 감사의 경우 이 점검은 감사가 실행된 마지막 시간부터 감사의 이 인스턴스가 시작된 시간까지의 데이터를 확인합니다. 점검된 시간 동안 이 상태를 완화하는 단계를 수행한 경우 문제가 지속되는지 여부를 확인하기 위해 연결/연결 해제가 발생한 시기를 기록해 둡니다.

이 점검에서 규정 미준수가 발견된 경우 다음 사유 코드가 반환됩니다.

- DUPLICATE_CLIENT_ID_ACROSS_CONNECTIONS

또한 이 점검에서 반환되는 결과에는 연결에 사용된 클라이언트 ID, 보안 주체 ID 및 연결 해제 횟수가 포함됩니다. 가장 최근 결과가 먼저 나열됩니다.

이것이 중요한 이유

ID가 충돌하는 디바이스는 끊임없이 다시 연결되므로 메시지가 손실되거나 디바이스가 연결할 수 없는 상태가 됩니다.

이는 디바이스 또는 디바이스의 자격 증명이 손상되었으며 DDoS 공격일 수 있음을 나타낼 수 있습니다. 또한 계정에서 디바이스가 올바르게 구성되지 않거나 디바이스의 연결이 잘못되어 1분당 여러 번 다시 시도될 수 있습니다.

수정 방법

각 디바이스를 AWS IoT에서 고유한 사물로 등록하고 사물 이름을 연결할 클라이언트 ID로 사용합니다. 또는 MQTT를 통해 디바이스를 연결할 때 UUID를 클라이언트 ID로 사용합니다. 완화 작업을 사용하면 다음을 수행할 수 있습니다.

- Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하세요.

디바이스 인증서 만료

디바이스 인증서가 구성된 기준값 기간 내에 만료되거나 이미 만료되었습니다. 인증서 만료 검사 기준값은 30일(최소)에서 3,652일(10년(최대)) 사이로 구성할 수 있으며 기본값은 30일입니다.

이 점검은 CLI 및 API에서 `DEVICE_CERTIFICATE_EXPIRING_CHECK`와(과) 같이 나타납니다.

심각도: 중간

세부 사항

이 점검은 ACTIVE 또는 PENDING_TRANSFER 상태인 디바이스 인증서에 적용됩니다.

이 점검에서 규정 미준수 디바이스 인증서가 발견된 경우 다음 사유 코드가 반환됩니다.

- CERTIFICATE_APPROACHING_EXPIRATION

- CERTIFICATE_PAST_EXPIRATION

이것이 중요한 이유

만료된 후에는 디바이스 인증서를 사용해서는 안 됩니다.

디바이스 인증서 만료 검사 구성

이 구성을 사용하면 디바이스 플릿 전반에서 만료 날짜가 다가오는 인증서에 대한 알림을 모니터링하고 수신할 수 있습니다. 예를 들어 인증서 만료 후 30일 이내에 알림을 받으려면 다음과 같이 검사를 구성할 수 있습니다.

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

수정 방법

방법은 보안 모범 사례를 참조하세요. 수행 가능한 작업은 다음과 같습니다.

1. 새 인증서를 프로비저닝하고 디바이스에 연결합니다.
2. 새 인증서가 유효하고 디바이스가 이 인증서를 사용하여 연결할 수 있는지 확인합니다.
3. UpdateCertificate를 사용하여 이전 인증서를 AWS IoT에서 비활성(INACTIVE)으로 표시합니다. 완화 작업을 사용하면 다음을 수행할 수 있습니다.
   - 이 변경사항을 실행하려면 감사 결과에서 `UPDATE_DEVICE_CERTIFICATE` 완화 작업을 적용합니다.
   - 조치를 취할 수 있는 그룹에 디바이스를 추가하려면 `ADD_THINGS_TO_THING_GROUP` 완화 조치를 적용합니다.

- Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 PUBLISH_FINDINGS_TO_SNS 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하세요.

4. 기존 인증서를 디바이스에서 분리합니다. (DetachThingPrincipal 참조)

디바이스 인증서 수명 확인

이 감사 확인은 디바이스 인증서가 지정한 수보다 크거나 동일한 일수 동안 활성화되어 있을 경우 알림을 전달합니다. 이 검사를 통해 인증서 상태에 대한 정보를 얻을 수 있는 덕분에 인증서 수명이 종료되더라도 주기적으로 적절한 조치를 취할 수 있어 인증서 손상 위험을 줄여 보안을 강화할 수 있습니다.

인증서 수명 검사 기준값은 30일(최소)에서 3,652일(10년(최대)) 사이로 구성할 수 있으며 기본값은 365일입니다.

이 점검은 CLI 및 API에서 DEVICE_CERTIFICATE_AGE_CHECK와(과) 같이 나타납니다. 이 검사는 기본적으로 심각도: 낮음으로 사용 해제되어 있습니다.

세부 사항

이 점검은 ACTIVE 또는 PENDING_TRANSFER 상태인 디바이스 인증서에 적용됩니다. 이 점검에서 규정 미준수 디바이스 인증서가 발견된 경우 다음 사유 코드가 반환됩니다.

- CERTIFICATE_PAST_AGE_THRESHOLD

디바이스 인증서 수명 확인 구성

이 구성을 사용하면 플릿의 특정 요구 사항에 맞게 인증서 교체 알림을 맞춤화하여 모든 디바이스 전반에서 강력한 보안 태세를 유지할 수 있습니다. UpdateAccountAuditConfiguration API를 사용하여 이 검사를 구성할 수 있습니다. 예를 들어 인증서가 365일 이상 활성 상태일 때 알림을 받으려면 다음과 같이 검사를 구성할 수 있습니다.

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

```
                "CERT_AGE_THRESHOLD_IN_DAYS": "365"
            }
        }
    }
}
```

취소된 디바이스 인증서가 계속 활성 상태

취소된 디바이스 인증서가 계속 활성 상태입니다.

이 점검은 CLI 및 API에서 `REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK`와(과) 같이 나타납니다.

심각도: 중간

세부 사항

디바이스 인증서가 해당 CA의 인증서 해지 목록에 있지만, AWS IoT에서 여전히 활성 상태입니다.

이 점검은 ACTIVE 또는 PENDING_TRANSFER 상태인 디바이스 인증서에 적용됩니다.

이 점검에서 규정 미준수가 발견된 경우 다음 사유 코드가 반환됩니다.

- CERTIFICATE_REVOKED_BY_ISSUER

이것이 중요한 이유

일반적으로 손상되었으므로 디바이스 인증서가 취소됩니다. 오류 또는 실수로 인해 AWS IoT에서 아직 취소되지 않을 수 있습니다.

수정 방법

디바이스 인증서가 손상되지 않았는지 확인합니다. 손상되었다면 보안 모범 사례를 따라 상황을 완화시킵니다. 수행 가능한 작업은 다음과 같습니다.

1. 디바이스에 새 인증서를 프로비저닝합니다.
2. 새 인증서가 유효하고 디바이스가 이 인증서를 사용하여 연결할 수 있는지 확인합니다.
3. UpdateCertificate을 사용하여 이전 인증서를 AWS IoT에서 취소됨(REVOKED)으로 표시합니다. 완화 작업을 사용하면 다음을 수행할 수 있습니다.

- 이 변경사항을 실행하려면 감사 결과에서 `UPDATE_DEVICE_CERTIFICATE` 완화 작업을 적용합니다.
- 조치를 취할 수 있는 그룹에 디바이스를 추가하려면 `ADD_THINGS_TO_THING_GROUP` 완화 조치를 적용합니다.
- Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하세요.

4. 기존 인증서를 디바이스에서 분리합니다. (DetachThingPrincipal 참조)

로깅 비활성화

AWS IoT 로그는 Amazon CloudWatch에서 활성화되지 않습니다. V1, V2 로깅을 모두 확인합니다.

이 점검은 CLI 및 API에서 `LOGGING_DISABLED_CHECK`와(과) 같이 나타납니다.

심각도: 낮음

세부 사항

이 점검에서 규정 미준수가 발견된 경우 다음 사유 코드가 반환됩니다.

- LOGGING_DISABLED

이것이 중요한 이유

CloudWatch의 AWS IoT 로그를 통해 인증 실패, 예상치 못한 연결 및 연결 해제 등 디바이스가 손상되었음을 나타낼 수 있는 AWS IoT 내의 동작을 파악할 수 있습니다.

수정 방법

CloudWatch의 AWS IoT 로그를 활성화합니다. AWS IoT Core 개발자 안내서의 모니터링 및 로깅 완화 작업을 사용하면 다음을 수행할 수 있습니다.

- 이 변경사항을 실행하려면 감사 결과에서 `ENABLE_IOT_LOGGING` 완화 작업을 적용합니다.
- Amazon SNS 메시지에 대해 사용자 지정 응답을 구현하려면 `PUBLISH_FINDINGS_TO_SNS` 완화 작업을 적용합니다.

자세한 내용은 완화 작업 단원을 참조하십시오.

감사 명령

감사 설정 관리

`UpdateAccountAuditConfiguration`을 사용하여 계정에 대한 감사 설정을 구성합니다. 이 명령을 사용하면 감사에 사용할 점검을 활성화하고, 선택적 알림을 설정하고, 권한을 구성할 수 있습니다.

`DescribeAccountAuditConfiguration`으로 이러한 설정을 점검합니다.

감사 설정을 삭제하려면 `DeleteAccountAuditConfiguration`을 사용합니다. 이렇게 하면 모든 기본값이 복원되며, 모든 점검이 기본적으로 비활성화되므로 효율적으로 감사를 비활성화합니다.

UpdateAccountAuditConfiguration

이 계정에 대한 Device Defender 감사 설정을 구성하거나 재구성합니다. 설정에는 감사 알림을 전송하는 방법과 활성화 또는 비활성화된 감사 검사가 포함되어 있습니다.

시놉시스

```
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` 형식

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

```
      "enabled": "boolean"
    }
  }
}
```

cli-input-json Fields

명칭	유형	설명
roleArn	문자열 최대 길이: 2048 최소 길이: 20	감사를 수행할 때 AWS IoT에 디바이스, 정책, 인증서 및 기타 항목에 대한 정보에 액세스할 수 있는 권한을 부여하는 역할의 ARN입니다.
auditNotificationTargetConfigurations	map	감사 알림을 전송할 대상에 대한 정보입니다.
targetArn	문자열	감사 알림을 전송할 대상(SNS 주제)의 ARN입니다.
roleArn	문자열 최대 길이: 2048 최소 길이: 20	대상으로 알림을 전송할 권한을 부여하는 역할의 ARN입니다.
활성화	boolean	대상에 대한 알림이 활성화된 경우 True입니다.
auditCheckConfigurations	map	이 계정에 대해 활성화 및 비활성화된 감사 검사를 지정합니다. 현재 활성화된 점검을 비롯한 모든 점검 목록을 보려면 `DescribeAccountAuditConfiguration` 을 사용합니다. 특정 점검이 활성화된 경우 일부 데이터 수집이 즉시 시작될 수 있습니다. 점검이 비활성화되면 점검과 관련하여 지금까

명칭	유형	설명
		지 수집된 모든 데이터가 삭제됩니다. 예정된 감사에서 사용되는 경우에는 점검을 비활성화할 수 없습니다. 먼저 예정된 감사에서 점검을 삭제하거나 예정된 감사 자체를 삭제해야 합니다. `UpdateAccountAuditConfiguration` 에 대한 첫 번째 호출에서 이 파라미터는 필수이며, 하나 이상의 활성화된 점검을 지정해야 합니다.
활성화	boolean	이 계정에 대해 이 감사 검사가 활성화된 경우 True입니다.
구성	map	(선택 사항) `CERT_AGE_THRESHOLD_IN_DAYS` 및 `CERT_EXPIRATION_THRESHOLD_IN_DAYS` 등의 특정 감사 검사에 대한 사용자 지정 구성을 사용하면 인증서 수명 및 임박한 만료에 대한 알림을 받을 시기를 정의할 수 있습니다.

출력

없음

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

DescribeAccountAuditConfiguration

이 계정에 대한 Device Defender Audit 설정에 대한 정보를 가져옵니다. 설정에는 감사 알림을 전송하는 방법과 활성화 또는 비활성화된 감사 검사가 포함되어 있습니다.

시놉시스

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

`cli-input-json` 형식

```
{
}
```

출력

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

CLI 출력 필드

명칭	유형	설명
roleArn	문자열 최대 길이: 2048 최소 길이: 20	감사를 수행할 때 AWS IoT에 디바이스, 정책, 인증서 및 기타 항목에 대한 정보에 액세스할 수 있는 권한을 부여하는 역할의 ARN입니다. `UpdateAccountAuditConfiguration` 에 대한 첫 번째 호출에서 이 파라미터는 필수입니다.
auditNotificationTargetConfigurations	map	이 계정의 감사 알림을 전송할 대상에 대한 정보입니다.
targetArn	문자열	감사 알림을 전송할 대상(SNS 주제)의 ARN입니다.
roleArn	문자열 최대 길이: 2048 최소 길이: 20	대상으로 알림을 전송할 권한을 부여하는 역할의 ARN입니다.
활성화	boolean	대상에 대한 알림이 활성화된 경우 True입니다.
auditCheckConfigurations	map	이 계정에 대해 활성화 및 비활성화된 감사 검사입니다.
활성화	boolean	이 계정에 대해 이 감사 검사가 활성화된 경우 True입니다.
구성	map	(선택 사항) 여기서는 인증서의 최대 허용 기간 또는 알림이 트리거되어야 하는 만료 전 일수와 같은 특정 감사 검사에 적합한 특정 구성을 제공합니다.

오류

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

DeleteAccountAuditConfiguration

이 계정에 대한 Device Defender 감사의 기본 설정을 복원합니다. 입력한 모든 구성 데이터가 삭제되고 모든 감사 검사가 비활성화됨으로 재설정됩니다.

시놉시스

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

`cli-input-json` 형식

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

`cli-input-json` Fields

명칭	유형	설명
deleteScheduledAudits	boolean	true인 경우 예정된 감사가 모두 삭제됩니다.

출력

없음

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ResourceNotFoundException`

지정한 리소스가 존재하지 않습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

감사 예약

`CreateScheduledAudit`을 사용하여 하나 이상의 예정된 감사를 생성합니다. 이 명령을 사용하면 감사 중 수행할 점검과 감사를 실행해야 하는 빈도를 지정할 수 있습니다.

`ListScheduledAudits` 및 `DescribeScheduledAudit`을 통해 예정된 감사를 계속 추적합니다.

`UpdateScheduledAudit`으로 기존 예정된 감사를 변경하거나 `DeleteScheduledAudit`으로 삭제합니다.

CreateScheduledAudit

지정된 시간 간격으로 실행되는 예정된 감사를 생성합니다.

시놉시스

```
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` 형식

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

cli-input-json Fields

명칭	유형	설명
frequency	문자열	예정된 감사가 발생하는 빈도입니다. DAILY, WEEKLY, BIWEEKLY 또는 MONTHLY 중 하나일 수 있습니다. 각 감사의 실제 시작 시간은 시스템에서 결정됩니다. 열거형: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	문자열 패턴: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	예정된 감사가 발생하는 월의 날입니다. 1~31 또는 LAST일 수 있습니다. `frequency` 파라미터가 MONTHLY로 설정된 경우 이 필드는 필수입니다. 29~31일이 지정되고 그만큼 많은 날이 없는 월의 경우에는 월의 LAST 날에 감사가 발생합니다.

명칭	유형	설명
dayOfWeek	문자열	예정된 감사가 발생하는 주의 날입니다. SUN, MON, TUE, WED, THU, FRI 또는 SAT 중 하나일 수 있습니다. `frequency` 파라미터가 WEEKLY 또는 BIWEEKLY로 설정된 경우 이 필드는 필수입니다. 열거형: SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	list 멤버: AuditCheckName	예정된 감사 중 어떤 점검이 수행되는지를 나타냅니다. 계정에 대해 점검이 활성화되어야 합니다. (활성화된 점검을 비롯한 모든 점검 목록을 보려면 `DescribeAccountAuditConfiguration` 을 사용하고, 활성화된 점검을 선택하려면 `UpdateAccountAuditConfiguration` 을 사용합니다.)
tags	list 멤버: Tag java 클래스: java.util.List	예정된 감사 관리에 사용할 수 있는 메타데이터입니다.
키	문자열	태그 키.
값	문자열	태그 값.

명칭	유형	설명
scheduledAuditName	문자열 최대 길이: 128 최소 길이: 1 패턴: [a-zA-Z0-9_-]+	예정된 감사에 제공할 이름입니다. (최대 128자)

출력

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

CLI 출력 필드

명칭	유형	설명
scheduledAuditArn	문자열	예정된 감사의 ARN입니다.

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

`LimitExceededException`

한도를 초과했습니다.

ListScheduledAudits

예정된 감사를 모두 나열합니다.

시놉시스

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

`cli-input-json` 형식

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

`cli-input-json` Fields

명칭	유형	설명
nextToken	문자열	다음 결과 집합에 대한 토큰입니다.
maxResults	정수 최대 범위: 250 최소 범위: 1	한 번에 반환할 수 있는 최대 결과 수입니다. 기본값은 25입니다.

출력

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

CLI 출력 필드

명칭	유형	설명
scheduledAudits	list 멤버: ScheduledAuditMetadata java 클래스: java.util.List	예정된 감사의 목록입니다.
scheduledAuditName	문자열 최대 길이: 128 최소 길이: 1 패턴: [a-zA-Z0-9_-]+	예약된 감사의 이름입니다.
scheduledAuditArn	문자열	예정된 감사의 ARN입니다.
frequency	문자열	예정된 감사가 발생하는 빈도입니다. 열거형: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	문자열 패턴: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	예정된 감사가 실행되는 월의 날입니다(`frequency` 가 MONTHLY인 경우). 29~31일이 지정되고 그만큼 많은 날이 없는 월의 경우에는 월의 LAST 날에 감사가 발생합니다.
dayOfWeek	문자열	예정된 감사가 실행되는 주의 날입니다(`frequency` 가 WEEKLY 또는 BIWEEKLY인 경우). 열거형: SUN | MON | TUE | WED | THU | FRI | SAT

명칭	유형	설명
nextToken	문자열	다음 결과 집합을 가져오는 데 사용할 수 있는 토큰 또는 결과가 더 이상 없는 경우 `null`입니다.

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

DescribeScheduledAudit

예정된 감사에 대한 정보를 가져옵니다.

시놉시스

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

`cli-input-json` 형식

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

cli-input-json Fields

명칭	유형	설명
scheduledAuditName	문자열 최대 길이: 128 최소 길이: 1 패턴: [a-zA-Z0-9_-]+	가져올 정보가 포함된 예정된 감사의 이름입니다.

출력

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

CLI 출력 필드

명칭	유형	설명
frequency	문자열	예정된 감사가 발생하는 빈도입니다. DAILY, WEEKLY, BIWEEKLY 또는 MONTHLY 중 하나입니다. 각 감사의 실제 시작 시간은 시스템에서 결정됩니다. 열거형: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	문자열	예정된 감사가 발생하는 월의 날입니다. 1~31 또는 LAST일 수 있습니다. 29~31일이 지정

명칭	유형	설명
	패턴: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	되고 그만큼 많은 날이 없는 월의 경우에는 월의 LAST 날에 감사가 발생합니다.
dayOfWeek	문자열	예정된 감사가 발생하는 주의 날입니다. SUN, MON, TUE, WED, THU, FRI 또는 SAT 중 하나입니다. 열거형: SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	list 멤버: AuditCheckName	예정된 감사 중 어떤 점검이 수행되는지를 나타냅니다. 계정에 대해 점검이 활성화되어야 합니다. (활성화된 점검을 비롯한 모든 점검 목록을 보려면 `DescribeAccountAuditConfiguration` 을 사용하고, 활성화된 점검을 선택하려면 `UpdateAccountAuditConfiguration` 을 사용합니다.)
scheduledAuditName	문자열 최대 길이: 128 최소 길이: 1 패턴: [a-zA-Z0-9_-]+	예약된 감사의 이름입니다.
scheduledAuditArn	문자열	예정된 감사의 ARN입니다.

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ResourceNotFoundException`

지정한 리소스가 존재하지 않습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

UpdateScheduledAudit

수행되는 점검 및 감사가 발생하는 빈도를 비롯하여 예정된 감사를 업데이트합니다.

시놉시스

```
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` 형식

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

`cli-input-json` Fields

명칭	유형	설명
frequency	문자열	예정된 감사가 발생하는 빈도입니다. DAILY, WEEKLY, BIWEEKLY 또는 MONTHLY 중 하나일 수 있습니다. 각 감사의 실제 시작 시간은 시스템에서 결정됩니다. 열거형: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	문자열 패턴: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	예정된 감사가 발생하는 월의 날입니다. 1~31 또는 LAST일 수 있습니다. `frequency` 파라미터가 MONTHLY로 설정된 경우 이 필드는 필수입니다. 29~31일이 지정되고 그만큼 많은 날이 없는 월의 경우에는 월의 LAST 날에 감사가 발생합니다.
dayOfWeek	문자열	예정된 감사가 발생하는 주의 날입니다. SUN, MON, TUE, WED, THU, FRI 또는 SAT 중 하나일 수 있습니다. `frequency` 파라미터가 WEEKLY 또는 BIWEEKLY로 설정된 경우 이 필드는 필수입니다. 열거형: SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	list 멤버: AuditCheckName	예정된 감사 중 어떤 점검이 수행되는지를 나타냅니다. 계정에 대해 점검이 활성화되어

명칭	유형	설명
		야 합니다. (활성화된 점검을 비롯한 모든 점검 목록을 보려면 `DescribeAccountAuditConfiguration` 을 사용하고, 활성화된 점검을 선택하려면 `UpdateAccountAuditConfiguration` 을 사용합니다.)
scheduledAuditName	문자열 최대 길이: 128 최소 길이: 1 패턴: [a-zA-Z0-9_-]+	예약된 감사의 이름입니다. (최대 128자)

출력

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

CLI 출력 필드

명칭	유형	설명
scheduledAuditArn	문자열	예정된 감사의 ARN입니다.

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ResourceNotFoundException`

지정한 리소스가 존재하지 않습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

DeleteScheduledAudit

예정된 감사를 삭제합니다.

시놉시스

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

`cli-input-json` 형식

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

`cli-input-json` Fields

명칭	유형	설명
scheduledAuditName	문자열 최대 길이: 128 최소 길이: 1 패턴: [a-zA-Z0-9_-]+	삭제할 예정된 감사의 이름입니다.

출력

없음

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ResourceNotFoundException`

지정한 리소스가 존재하지 않습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

온디맨드 감사 실행

`StartOnDemandAuditTask`를 사용하여 수행할 점검을 지정하고 즉시 감사를 실행합니다.

StartOnDemandAuditTask

온디맨드 Device Defender 감사를 시작합니다.

시놉시스

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

`cli-input-json` 형식

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

`cli-input-json` Fields

명칭	유형	설명
targetCheckNames	list 멤버: AuditCheckName	감사 중 어떤 점검이 수행되는지를 나타냅니다. 지정하는 점검은 계정에 대해 활성화되어야 합니다. 그렇지 않으면 예외가 발생합니다. 활성화된 점검을 비롯한 모든 점검 목록을 보려면 `DescribeAccountAuditConfiguration` 을 사용하고, 활성화된 점검을 선택하려면 `UpdateAccountAuditConfiguration` 을 사용합니다.

출력

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

CLI 출력 필드

명칭	유형	설명
taskId	문자열 최대 길이: 40 최소 길이: 1 패턴: [a-zA-Z0-9-]+	시작한 온디맨드 감사의 ID입니다.

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

`LimitExceededException`

한도를 초과했습니다.

감사 인스턴스 관리

특정 감사 인스턴스에 대한 정보를 가져오려면 `DescribeAuditTask`를 사용합니다. 이미 실행된 경우 결과에는 실패한 점검과 통과된 점검, 시스템이 완료할 수 없는 점검 및 감사가 계속 진행 중인 경우 작업 중인 점검이 포함됩니다.

지정된 시간 간격 동안 실행된 감사를 찾으려면 `ListAuditTasks`를 사용합니다.

진행 중인 감사를 중지하려면 `CancelAuditTask`를 사용합니다.

DescribeAuditTask

Device Defender 감사에 대한 정보를 가져옵니다.

시놉시스

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

`cli-input-json` 형식

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

cli-input-json Fields

명칭	유형	설명
taskId	문자열 최대 길이: 40 최소 길이: 1 패턴: [a-zA-Z0-9-]+	가져올 정보가 포함된 감사의 ID입니다.

출력

```
{
  "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 출력 필드

명칭	유형	설명
taskStatus	문자열	감사의 상태로 IN_PROGRESS, COMPLETED, FAILED 또는 CANCELED 중 하나입니다. 열거형: IN_PROGRESS | COMPLETED | FAILED | CANCELED
taskType	문자열	감사 유형으로 ON_DEMAND_AUDIT_TASK 또는 SCHEDULED_AUDIT_TASK 입니다. 열거형: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStartTime	타임스탬프	감사가 시작된 시간입니다.
taskStatistics	TaskStatistics	감사에 대한 통계 정보입니다.
totalChecks	정수	이 감사의 점검 수입니다.
inProgressChecks	정수	진행 중인 점검 수입니다.
waitingForDataCollectionChecks	정수	데이터 수집 대기 중인 점검 수입니다.
compliantChecks	정수	규정 준수 리소스를 발견한 점검 수입니다.
nonCompliantChecks	정수	규정 미준수 리소스를 발견한 점검 수입니다.
failedChecks	정수	점검 수입니다.

명칭	유형	설명
canceledChecks	정수	감사가 취소되어 실행되지 않은 점검 수입니다.
scheduledAuditName	문자열 최대 길이: 128 최소 길이: 1 패턴: [a-zA-Z0-9_-]+	예정된 감사의 이름입니다(감사가 예정된 감사인 경우에만 해당).
auditDetails	map	이 감사 중 수행된 각 점검에 대한 세부 정보입니다.
checkRunStatus	문자열	이 점검의 완료 상태로 IN_PROGRESS, WAITING_FOR_DATA_COLLECTION, CANCELED, COMPLETED_COMPLIANT, COMPLETED_NON_COMPLIANT 또는 FAILED 중 하나입니다. 열거형: IN_PROGRESS | WAITING_FOR_DATA_COLLECTION | CANCELED | COMPLETED_COMPLIANT | COMPLETED_NON_COMPLIANT | FAILED
checkCompliant	boolean	점검이 완료되고 모든 리소스가 규정 준수로 확인된 경우 True입니다.
totalResourcesCount	long	점검이 수행된 리소스의 수입니다.
nonCompliantResourcesCount	long	점검에서 규정 미준수가 발견된 리소스의 수입니다.

명칭	유형	설명
errorCode	문자열	이 감사 중 이 점검을 수행할 때 발생한 오류 코드입니다. INSUFFICIENT_PERMISSIONS 또는 AUDIT_CHECK_DISABLED 중 하나입니다.
message	문자열 최대 길이: 2048	이 감사 중 이 점검을 수행할 때 발생한 오류와 연관된 메시지입니다.

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ResourceNotFoundException`

지정한 리소스가 존재하지 않습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

ListAuditTasks

지정된 기간 동안 수행된 Device Defender 감사를 나열합니다.

시놉시스

```
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 형식

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

cli-input-json Fields

명칭	유형	설명
startTime	타임스탬프	기간의 시작입니다. 감사 정보는 제한된 시간(180일) 동안 보관됩니다. 보관되기 전에 시작 시간을 요청하면 `InvalidRequestException` 이 발생합니다.
endTime	타임스탬프	기간의 끝입니다.
taskType	문자열	출력을 지정된 감사 유형으로 제한하기 위한 필터로 ON_DEMAND_AUDIT_TASK 또는 SCHEDULED__AUDIT_TASK 중 하나일 수 있습니다. 열거형: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStatus	문자열	감사에 대한 출력을 지정된 완료 상태로 제한하기 위

명칭	유형	설명
		한 필터로 IN_PROGRESS, COMPLETED, FAILED 또는 CANCELED 중 하나일 수 있습니다. 열거형: IN_PROGRESS | COMPLETED | FAILED | CANCELED
nextToken	문자열	다음 결과 집합에 대한 토큰입니다.
maxResults	정수 최대 범위: 250 최소 범위: 1	한 번에 반환할 수 있는 최대 결과 수입니다. 기본값은 25입니다.

출력

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

CLI 출력 필드

명칭	유형	설명
작업	list 멤버: AuditTaskMetadata java 클래스: java.util.List	지정된 기간 동안 수행된 감사입니다.

명칭	유형	설명
taskId	문자열 최대 길이: 40 최소 길이: 1 패턴: [a-zA-Z0-9-]+	이 감사의 ID입니다.
taskStatus	문자열	이 감사의 상태로 IN_PROGRESS, COMPLETED, FAILED 또는 CANCELED 중 하나입니다. 열거형: IN_PROGRESS | COMPLETED | FAILED | CANCELED
taskType	문자열	이 감사 유형으로 ON_DEMAND_AUDIT_TASK 또는 SCHEDULED_AUDIT_TASK 중 하나입니다. 열거형: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
nextToken	문자열	다음 결과 집합을 가져오기 위해 사용할 수 있는 토큰이지만 결과가 더 이상 없는 경우에는 `null`입니다.

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

CancelAuditTask

진행 중인 감사를 취소합니다. 감사는 예정된 감사이거나 온디맨드 감사일 수 있습니다. 감사가 진행 중이 아닌 경우 `InvalidRequestException`이 발생합니다.

시놉시스

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

`cli-input-json` 형식

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

`cli-input-json` Fields

명칭	유형	설명
taskId	문자열 최대 길이: 40 최소 길이: 1 패턴: [a-zA-Z0-9-]+	취소할 감사의 ID입니다. IN_PROGRESS인 감사만 취소할 수 있습니다.

출력

없음

오류

`ResourceNotFoundException`

지정한 리소스가 존재하지 않습니다.

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

감사 결과 점검

감사 결과를 확인하려면 `ListAuditFindings`를 사용합니다. 점검 유형, 특정 리소스 또는 감사 시간을 기준으로 결과를 필터링할 수 있습니다. 이 정보를 사용하여 발견된 문제를 완화할 수 있습니다.

완화 조치를 정의하고 감사 결과에 적용할 수 있습니다. 자세한 내용은 완화 작업 단원을 참조하세요.

ListAuditFindings

지정된 기간 동안 수행된 Device Defender 감사의 결과를 나열합니다. (결과는 180일 동안 보관됩니다.)

시놉시스

```
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` 형식

```
{
  "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 Fields

명칭	유형	설명
taskId	문자열 최대 길이: 40 최소 길이: 1 패턴: [a-zA-Z0-9-]+	감사에 대한 결과를 지정된 ID로 제한하는 필터입니다. taskId나 startTime 및 endTime을 지정해야 하지만, 둘 다 지정하면 안 됩니다.
checkName	문자열	결과를 지정된 감사 검사에 대한 결과로 제한하는 필터입니다.
resourceIdentifier	ResourceIdentifier	규정 미준수 리소스를 식별하는 정보입니다.
deviceCertificateId	문자열 최대 길이: 64 최소 길이: 64 패턴: (0x)?[a-fA-F0-9]+	리소스에 연결된 인증서의 ID입니다.
caCertificateId	문자열	인증서를 승인하는 데 사용되는 CA 인증서의 ID입니다.

명칭	유형	설명
	최대 길이: 64 최소 길이: 64 패턴: (0x)?[a-fA-F0-9]+	
cognitoIdentityPoolId	문자열	Amazon Cognito 자격 증명 풀의 ID입니다.
clientId	문자열	클라이언트 ID입니다.
policyVersionIdentifier	PolicyVersionIdentifier	리소스와 연관된 정책의 버전입니다.
policyName	문자열 최대 길이: 128 최소 길이: 1 패턴: [w+=,.@-]+	정책의 이름입니다.
policyVersionId	문자열 패턴: [0-9]+	리소스와 연관된 정책의 버전 ID입니다.
roleAliasArn	문자열	과도하게 허용적인 작업이 있는 역할 별칭의 ARN입니다. 최대 길이: 2048 최소 길이: 1
account	문자열 최대 길이: 12 최소 길이: 12 패턴: [0-9]+	리소스와 연관된 계정입니다.
maxResults	정수 최대 범위: 250 최소 범위: 1	한 번에 반환할 수 있는 최대 결과 수입니다. 기본값은 25입니다.
nextToken	문자열	다음 결과 집합에 대한 토큰입니다.

명칭	유형	설명
startTime	타임스탬프	결과를 지정된 시간 이후에 발견된 결과로 제한하는 필터입니다. startTime 및 endTime이나 taskId를 지정해야 하지만, 둘 다 지정하면 안 됩니다.
endTime	타임스탬프	결과를 지정된 시간 이전에 발견된 결과로 제한하는 필터입니다. startTime 및 endTime이나 taskId를 지정해야 하지만, 둘 다 지정하면 안 됩니다.

출력

```
{
  "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 출력 필드

명칭	유형	설명
결과	list 멤버: AuditFinding	감사의 결과입니다.
taskId	문자열	이 결과가 생성된 감사의 ID입니다.

명칭	유형	설명
	최대 길이: 40 최소 길이: 1 패턴: [a-zA-Z0-9-]+	
checkName	문자열	이 결과가 생성된 감사 검사입니다.
taskStartTime	타임스탬프	감사가 시작된 시간입니다.
findingTime	타임스탬프	결과가 발견된 시간입니다.
severity	문자열	결과의 심각도입니다. 열거형: CRITICAL | HIGH | MEDIUM | LOW
nonCompliantResource	NonCompliantResource	감사 검사를 통해 규정 미준수로 확인된 리소스입니다.
resourceType	문자열	규정 미준수 리소스의 유형입니다. 열거형: DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	규정 미준수 리소스를 식별하는 정보입니다.
deviceCertificateId	문자열 최대 길이: 64 최소 길이: 64 패턴: (0x)?[a-fA-F0-9]+	리소스에 연결된 인증서의 ID입니다.

명칭	유형	설명
caCertificateId	문자열 최대 길이: 64 최소 길이: 64 패턴: (0x)?[a-fA-F0-9]+	인증서를 승인하는 데 사용되는 CA 인증서의 ID입니다.
cognitoIdentityPoolId	문자열	Amazon Cognito 자격 증명 풀의 ID입니다.
clientId	문자열	클라이언트 ID입니다.
policyVersionIdentifier	PolicyVersionIdentifier	리소스와 연관된 정책의 버전입니다.
policyName	문자열 최대 길이: 128 최소 길이: 1 패턴: [w+=,.@-]+	정책의 이름입니다.
policyVersionId	문자열 패턴: [0-9]+	리소스와 연관된 정책의 버전 ID입니다.
account	문자열 최대 길이: 12 최소 길이: 12 패턴: [0-9]+	리소스와 연관된 계정입니다.
additionalInfo	map	규정 미준수 리소스에 대한 기타 정보입니다.
relatedResources	list 멤버: RelatedResource	관련 리소스의 목록입니다.

명칭	유형	설명
resourceType	문자열	리소스의 유형입니다. 열거형: DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	리소스를 식별하는 정보입니다.
deviceCertificateId	문자열 최대 길이: 64 최소 길이: 64 패턴: (0x)?[a-fA-F0-9]+	리소스에 연결된 인증서의 ID입니다.
caCertificateId	문자열 최대 길이: 64 최소 길이: 64 패턴: (0x)?[a-fA-F0-9]+	인증서를 승인하는 데 사용되는 CA 인증서의 ID입니다.
cognitoIdentityPoolId	문자열	Amazon Cognito 자격 증명 풀의 ID입니다.
clientId	문자열	클라이언트 ID입니다.
policyVersionIdentifier	PolicyVersionIdentifier	리소스와 연관된 정책의 버전입니다.
iamRoleArn	문자열 최대 길이: 2048 최소 길이: 20	과도하게 허용적인 작업이 있는 IAM 역할의 ARN입니다.

명칭	유형	설명
policyName	문자열 최대 길이: 128 최소 길이: 1 패턴: [w+=,.@-]+	정책의 이름입니다.
policyVersionId	문자열 패턴: [0-9]+	리소스와 연관된 정책의 버전 ID입니다.
roleAliasArn	문자열 최대 길이: 2048 최소 길이: 1	과도하게 허용적인 작업이 있는 역할 별칭의 ARN입니다.
account	문자열 최대 길이: 12 최소 길이: 12 패턴: [0-9]+	리소스와 연관된 계정입니다.
additionalInfo	map	리소스에 대한 기타 정보입니다.
reasonForNonCompliance	문자열	리소스가 규정 미준수인 사유입니다.
reasonForNonCompliance Code	문자열	리소스가 규정 미준수인 사유를 나타내는 코드입니다.
nextToken	문자열	다음 결과 집합을 가져오기 위해 사용할 수 있는 토큰이지만 결과가 더 이상 없는 경우에는 `null`입니다.

오류

`InvalidRequestException`

요청 내용이 잘못되었습니다.

`ThrottlingException`

속도가 제한을 초과했습니다.

`InternalFailureException`

예상치 못한 오류가 발생했습니다.

감사 결과 금지

감사를 실행하면 모든 비준수 리소스에 대한 검색 결과를 보고합니다. 즉, 감사 보고서에는 문제 완화를 위해 작업하고 있는 리소스와 테스트 또는 고장난 디바이스와 같이 규정을 준수하지 않는 것으로 알려진 리소스에 대한 결과가 포함됩니다. 감사는 연속적인 감사 실행에서 규정을 준수하지 않는 리소스에 대한 결과를 계속 보고하며, 이로 인해 원치 않는 정보가 보고서에 추가될 수 있습니다. 감사 결과 금지를 사용하면 리소스가 수정될 때까지 정의된 기간 동안 또는 테스트 또는 손상된 디바이스와 연결된 리소스에 대해 무기한으로 결과를 표시하지 않거나 필터링할 수 있습니다.

> **Note**
>
> 금지된 감사 결과에 대해서는 완화 작업을 사용할 수 없습니다. 완화 작업에 대한 자세한 내용은 완화 작업 단원을 참조하세요.

감사 결과 금지 할당량에 대한 자세한 내용은 AWS IoT Device Defender 엔드포인트 및 할당량을 참조하세요.

감사 결과 금지 작동 방식

비준수 리소스에 대한 감사 결과 금지를 만들면 감사 보고서와 알림이 다르게 작동합니다.

감사 보고서에는 보고서와 관련된 모든 금지된 결과를 나열하는 새 섹션이 포함됩니다. 감사 검사가 준수되는지 여부를 평가할 때 금지된 결과는 고려되지 않습니다. 명령줄 인터페이스(CLI)에서 describe-audit-task 명령을 사용하면 각 감사 검사에 대해 금지된 리소스 수도 반환됩니다.

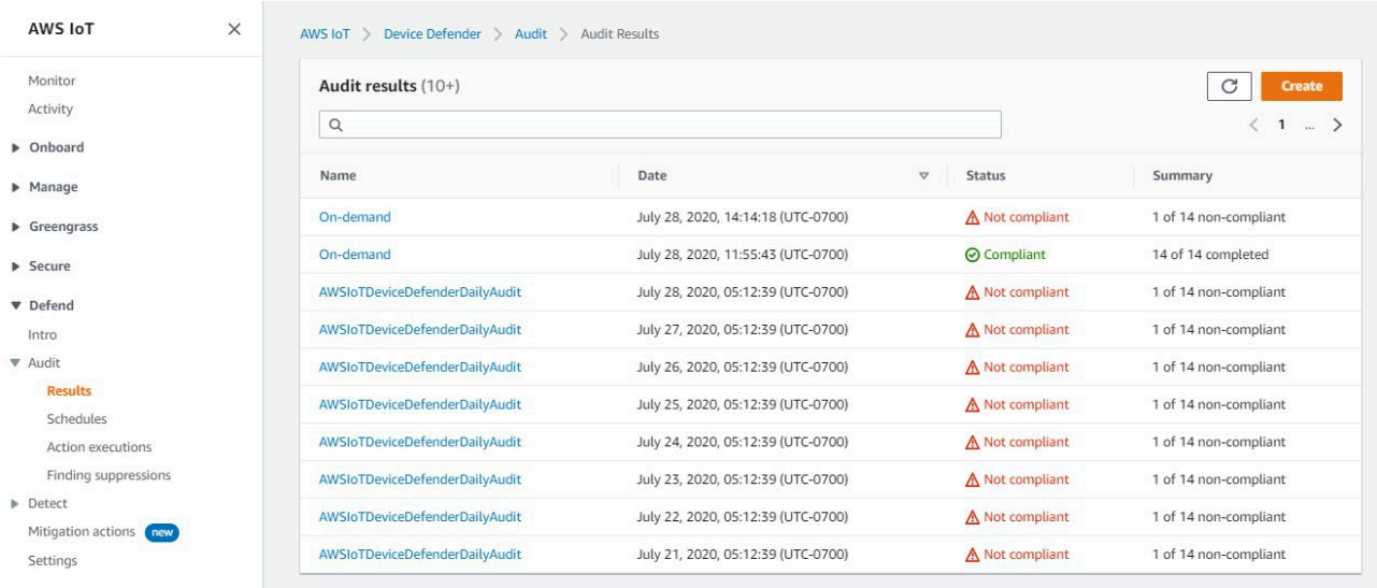
감사 알림의 경우, 감사 검사가 준수되는지 여부를 평가할 때 금지된 결과는 고려되지 않습니다. 금지된 리소스 수도 AWS IoT Device Defender이(가) Amazon CloudWatch 및 Amazon Simple Notification Service(Amazon SNS)에 게시하는 각 감사 검사 알림에 포함됩니다.

콘솔에서 감사 검사 금지를 사용하는 방법

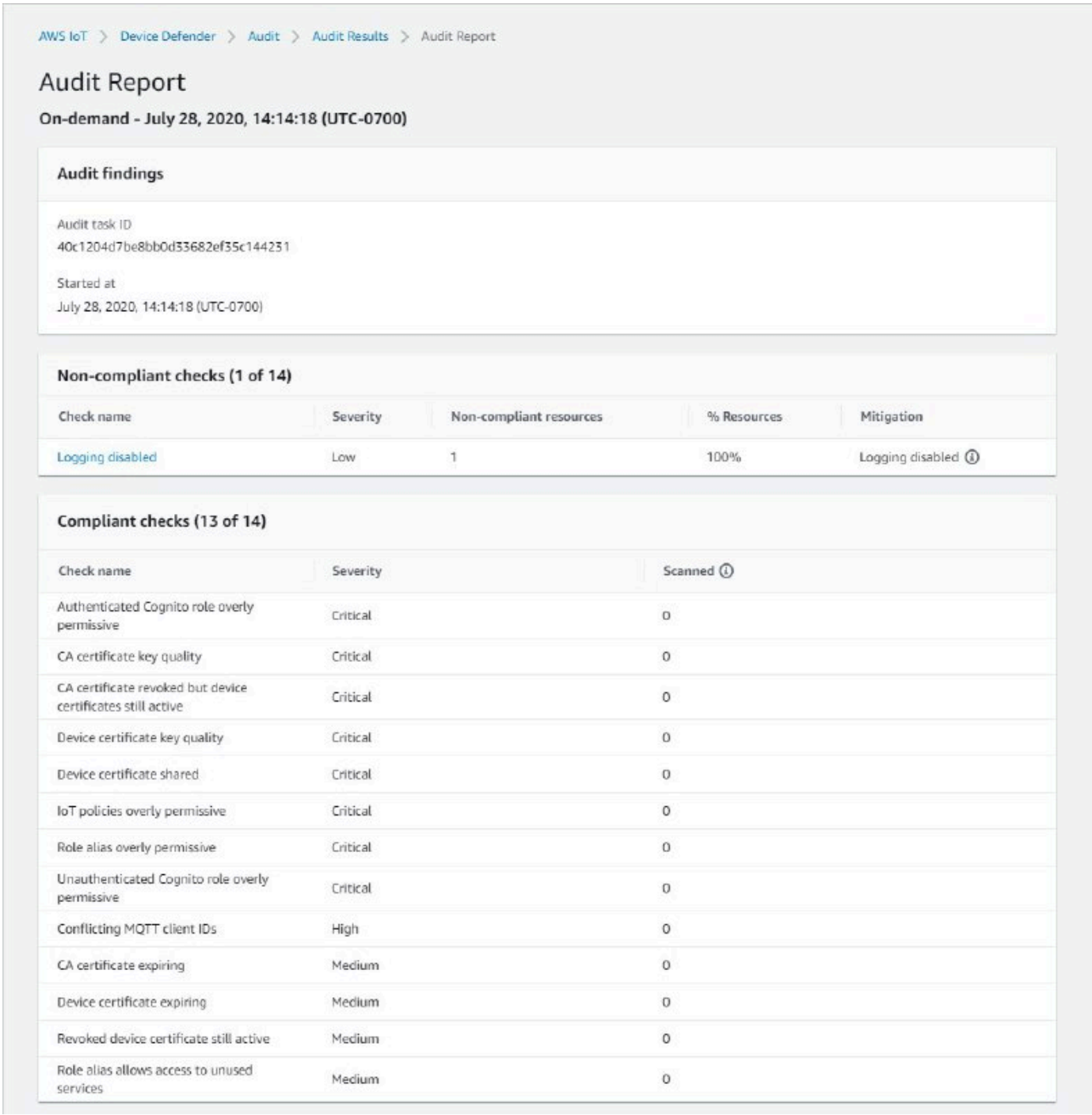
감사 보고서에서 결과를 금지하려면

다음 절차에서는 AWS IoT 콘솔에서 감사 결과 금지를 생성하는 방법을 소개합니다.

1. AWS IoT 콘솔의 탐색 창에서 방어를 확장하여 감사 및 결과를 차례로 선택합니다.
2. 검토하려는 감사 보고서를 선택합니다.



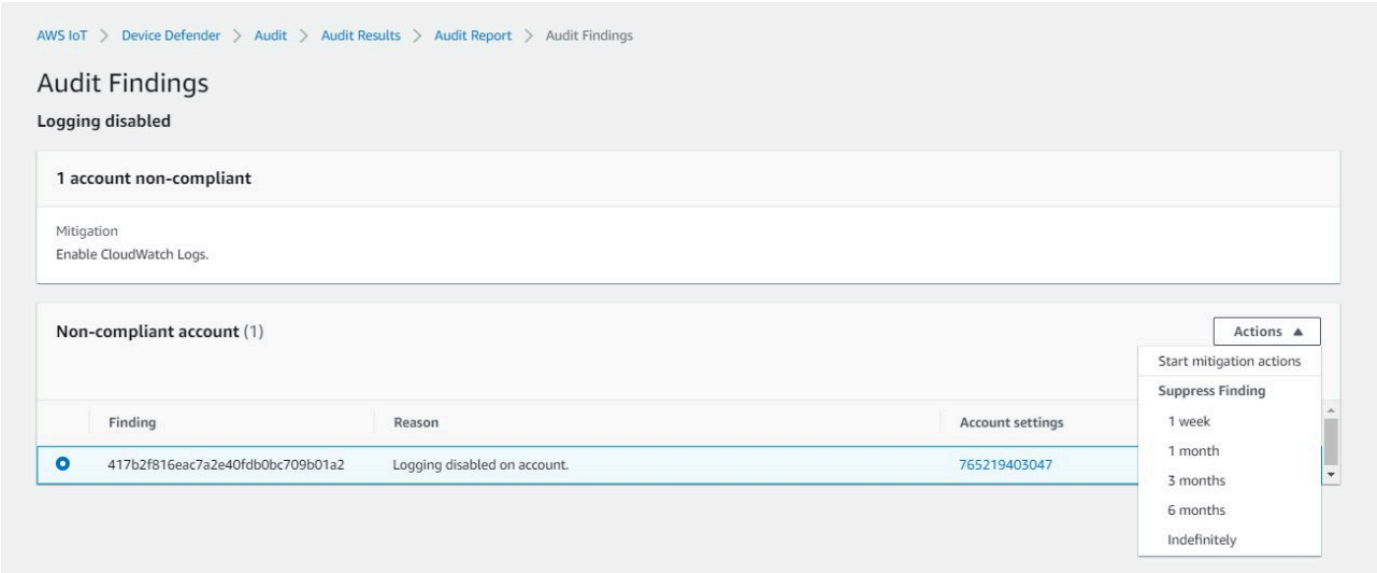
3. 미준수 점검 섹션의 이름 확인에서 관심 있는 감사 검사를 선택합니다.



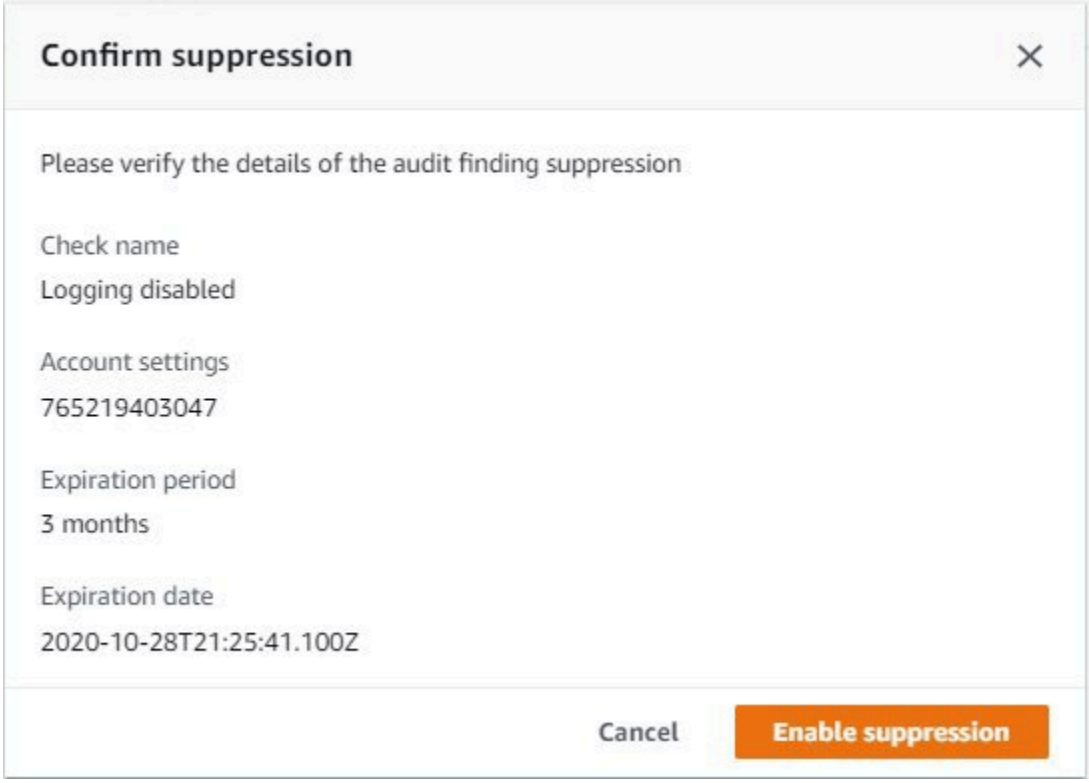
4. 감사 검사 세부 정보 화면에서 확인하지 않으려는 결과가 있는 경우 결과 옆에 있는 옵션 버튼을 선택합니다. 그런 다음 작업을 선택한 다음 감사 결과 금지를 지속할 시간을 선택합니다.

> **Note**
>
> 콘솔에서 1주(1 week), 1개월(1 month), 3개월(3 months), 6개월(6 months) 또는 무기한(Indefinitely)을 감사 결과 금지 만료 날짜로 선택할 수 있습니다. 특정 만료 날짜는 CLI 또

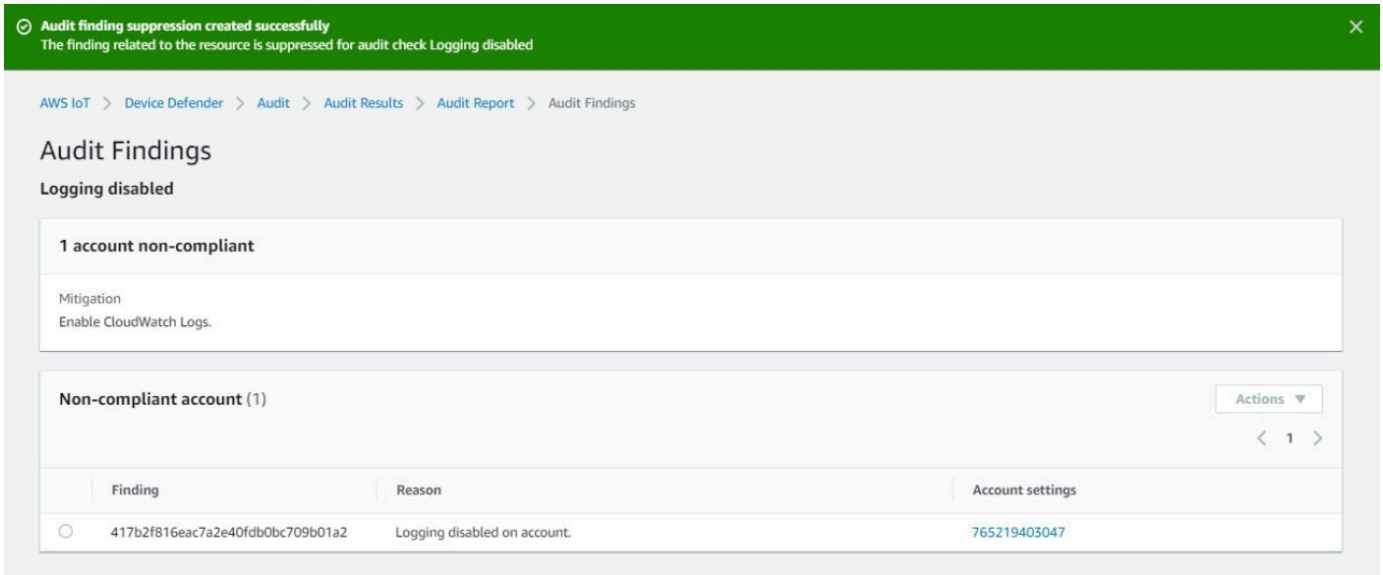
는 API에서만 설정할 수 있습니다. 감사 결과 금지는 만료 날짜에 관계없이 언제든지 취소할 수 있습니다.



5. 금지 세부 정보를 확인한 다음 금지 활성화를 선택합니다.

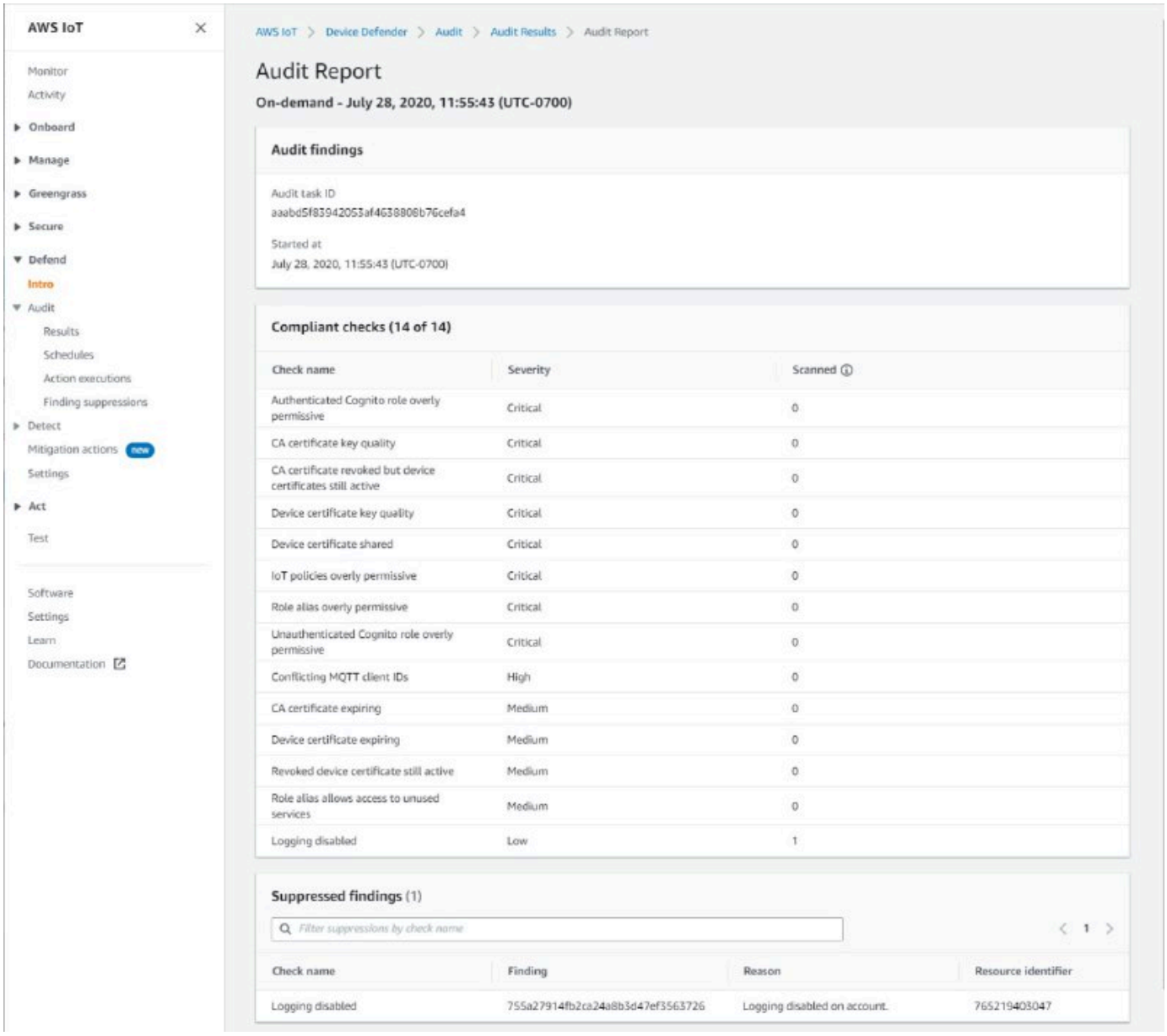


6. 감사 결과 금지를 생성한 후에는 감사 결과 금지가 만들어졌음을 확인하는 배너가 나타납니다.



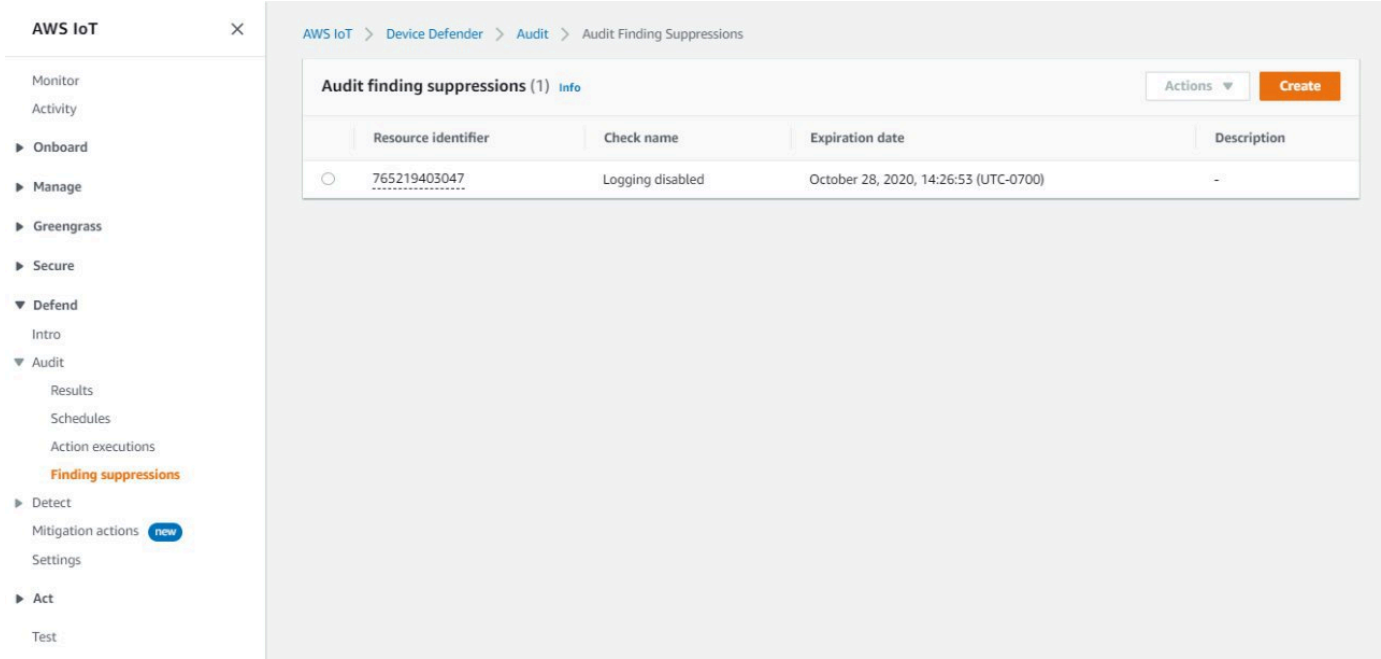
감사 보고서에서 금지된 결과를 보려면

1. AWS IoT 콘솔의 탐색 창에서 방어를 확장하여 감사 및 결과를 차례로 선택합니다.
2. 검토하려는 감사 보고서를 선택합니다.
3. 금지된 결과 섹션에서 선택한 감사 보고서에 대해 어떤 감사 결과가 금지되는지 확인합니다.



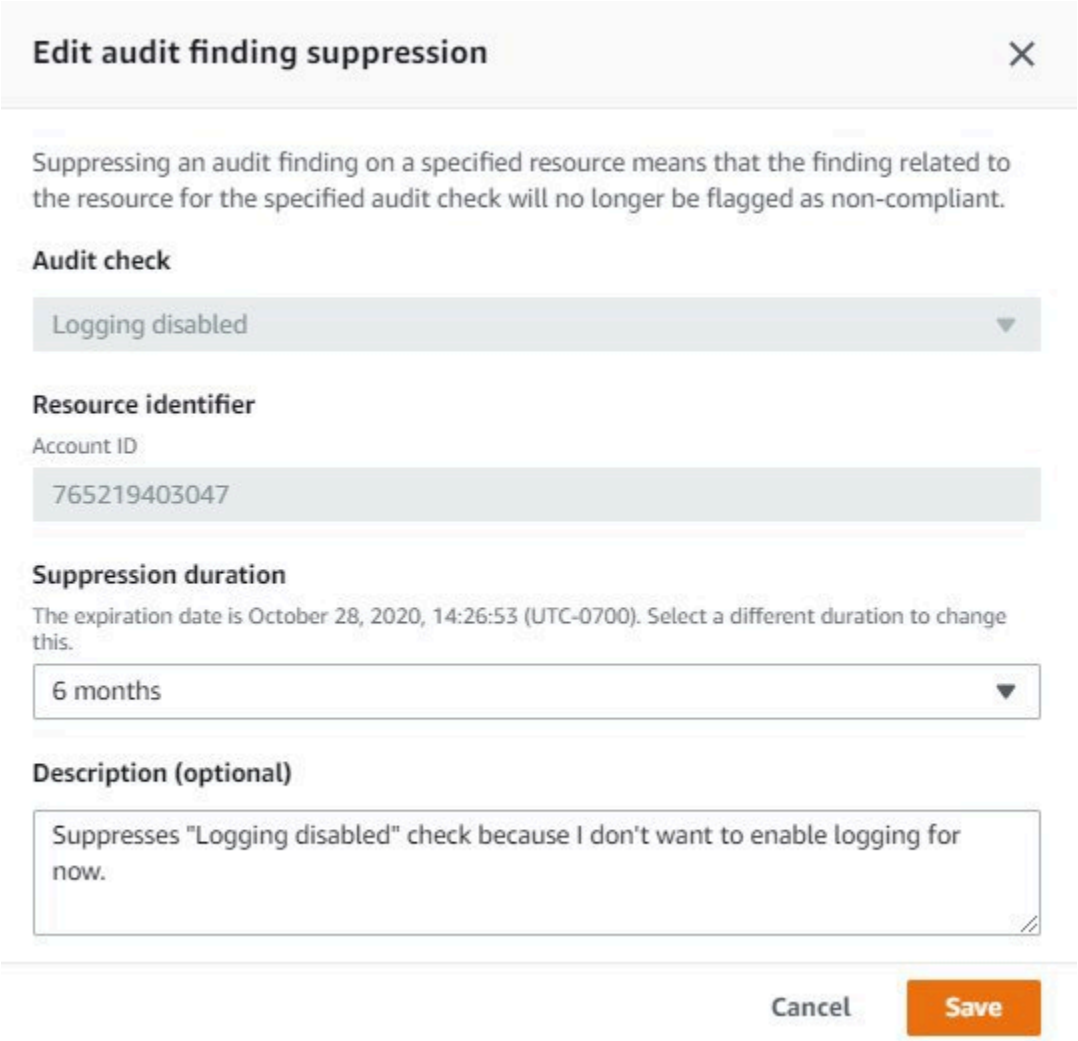
감사 결과 금지를 나열하려면

- AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 결과 금지를 차례로 선택합니다.



감사 결과 금지를 편집하려면

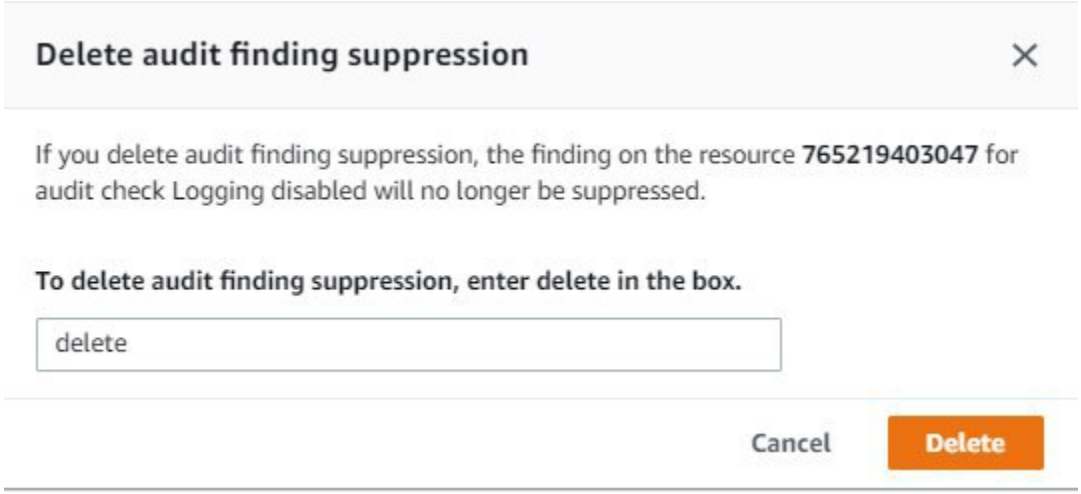
1. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 결과 금지를 차례로 선택합니다.
2. 편집하려는 감사 결과 금지 옆에 있는 옵션 버튼을 선택합니다. 다음으로, 작업(Actions), 편집(Edit)을 선택합니다.
3. 감사 결과 금지 편집 창에서 금지 기간 또는 설명(선택 사항)을 변경할 수 있습니다.



4. 변경을 적용한 후 저장을 선택합니다. 결과 금지 창이 열립니다.

감사 결과 금지를 삭제하려면

1. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 결과 금지를 차례로 선택합니다.
2. 삭제하려는 감사 결과 금지 옆에 있는 옵션 버튼을 선택한 후작업, 삭제를 차례로 선택합니다.
3. 감사 결과 금지 삭제 창에서 삭제를 확인할 입력란에 `delete`을(를) 입력한 다음 삭제를 선택합니다. 결과 금지 창이 열립니다.



CLI에서 감사 결과 금지를 사용하는 방법

다음 CLI 명령을 사용하여 감사 결과 금지를 만들고 관리할 수 있습니다.

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

입력한 `resource-identifier`은(는) 결과를 금지하는 `check-name`에 따라 달라집니다. 다음 표는 금지 생성 및 편집을 위해 어떤 검사가 어떤 `resource-identifier`을(를) 필요로 하는지 자세히 보여줍니다.

> **Note**
> 금지 명령은 감사를 비활성화하는 것을 나타내지 않습니다. 감사는 여전히 AWS IoT 디바이스에서 실행됩니다. 금지는 감사 결과에만 적용할 수 있습니다.

check-name	resource-identifier
AUTHENTICATE_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	cognitoIdentityPoolId

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

감사 결과 금지를 만들고 적용하려면

다음 절차에서는 AWS CLI에서 감사 결과 금지를 생성하는 방법을 소개합니다.

- `create-audit-suppression` 명령을 사용하여 감사 결과 금지를 생성합니다. 다음 예제에서는 로깅 비활성화 검사를 기반으로 AWS 계정 *123456789012*에 대해 감사 결과 금지를 생성합니다.

```
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."
```

  이 명령에 대한 출력이 없습니다.

감사 결과 금지 API

다음 API를 사용하여 감사 결과 금지를 만들고 관리할 수 있습니다.

- CreateAuditSuppression
- DescribeAuditSuppression
- UpdateAuditSuppression
- DeleteAuditSuppression
- ListAuditSuppressions

특정 감사 결과에 대해 필터링하려면 ListAuditFindings API를 사용할 수 있습니다.

감지

AWS IoT Device Defender Detect를 사용하면 디바이스의 동작을 모니터링하여 손상된 디바이스를 나타낼 수 있는 비정상적인 동작을 식별할 수 있습니다. 클라우드 측 지표(AWS IoT)와 디바이스 측 지표(디바이스에 설치한 에이전트)를 조합하여 사용하면 다음을 감지할 수 있습니다.

- 연결 패턴의 변경 사항.
- 권한이 없거나 인식할 수 없는 엔드포인트와 통신하는 디바이스입니다.
- 인바운드 및 아웃바운드 디바이스 트래픽 패턴의 변경 사항.

예상된 디바이스 동작의 정의가 포함된 보안 프로파일을 생성하고 이 프로파일을 디바이스 그룹이나 플릿의 모든 디바이스에 할당합니다. AWS IoT Device Defender Detect는 이러한 보안 프로파일을 사용하여 변칙을 감지하고 Amazon CloudWatch 지표 및 Amazon Simple Notification Service 알림을 통해 경보를 전송합니다.

AWS IoT Device Defender Detect를 통해 연결된 디바이스에서 자주 발견되는 보안 문제를 감지할 수 있습니다.

- 디바이스에서 잠재적인 악성 명령과 제어 채널을 나타내는 권한 없는 엔드포인트 또는 알려진 악의적인 IP 주소로의 트래픽
- 아웃바운드 트래픽의 급증과 같이 디바이스가 DDoS에 동원되고 있음을 나타내는 변칙적 트래픽
- 원격으로 액세스할 수 있는 원격 관리 인터페이스와 포트가 있는 디바이스
- 계정에 전송된 메시지 속도의 급증(예: 메시지당 요금이 과도하게 부과될 수 있는 악성 디바이스의 메시지인 경우).

사용 사례:

공격 대상 영역 측정

AWS IoT Device Defender Detect를 사용하여 디바이스의 공격 대상 영역을 측정할 수 있습니다. 예를 들어, 종종 공격 캠페인의 대상이 되는 서비스 포트가 있는 디바이스를 식별할 수 있습니다(포트 23/2323에서 실행되는 telnet 서비스, 포트 22에서 실행되는 SSH 서비스, 포트 80/443/8080/8081에서 실행되는 HTTP/S 서비스). 이러한 서비스 포트가 디바이스에서 사용될 합법적인 이유가 있는 경우에도 해당 포트는 또한 일반적으로 공격자가 공격 대상으로 하는 영역의 일부이며 관련된 위험을 수반하기도 합니다. AWS IoT Device Defender Detect에서 공격 대상 영역

을 알려준 경우 해당 영역을 최소화(미사용 네트워크 서비스 제거)하거나, 추가 평가를 실행하여 보안 취약점(예: 일반, 기본 또는 약한 암호로 구성된 telnet)을 식별합니다.

디바이스 동작의 변칙 감지 및 가능한 보안 근본 원인 파악

AWS IoT Device Defender Detect를 사용하여 보안 위반을 나타낼 수도 있는 예상치 못한 디바이스 동작 지표(개방된 포트의 수, 연결 수, 예상치 못한 개방 포트, 예상치 못한 IP 주소로의 연결)에 대한 경보를 받을 수 있습니다. 예를 들어, 예상 TCP 연결 수보다 더 높은 수는 디바이스가 DDoS 공격에 사용되고 있음을 나타낼 수 있습니다. 예상한 포트 이외의 포트에서 수신하는 프로세스는 원격 제어를 위해 디바이스에 설치된 백도어를 나타낼 수 있습니다. AWS IoT Device Defender Detect를 사용하여 디바이스 플릿의 상태를 프로브하고 보안 추정(예: 포트 23 또는 2323에서 수신하는 디바이스 없음)을 확인합니다.

기계 학습(ML) 기반 위협 감지를 활성화하여 잠재적인 위협을 자동으로 식별할 수 있습니다.

잘못 구성된 디바이스 감지

디바이스에서 계정으로 전송된 메시지의 수 또는 크기가 급증하는 것은 잘못 구성된 디바이스를 나타낼 수 있습니다. 이러한 디바이스는 메시지당 요금을 증가시킬 수 있습니다. 마찬가지로, 권한 부여가 여러 번 실패한 디바이스에는 재구성된 정책이 필요할 수 있습니다.

등록되지 않은 디바이스의 동작 모니터링

AWS IoT Device Defender Detect를 사용하면 AWS IoT 레지스트리에 등록되지 않은 디바이스의 비정상적인 동작을 식별할 수 있습니다. 다음 대상 유형 중 하나와 관련된 보안 프로파일을 정의할 수 있습니다.

- 모든 디바이스
- 등록된 모든 디바이스(AWS IoT 레지스트리의 사물)
- 등록되지 않은 모든 디바이스
- 사물 그룹에 속한 디바이스

보안 프로파일은 계정의 디바이스에 대해 예상한 동작을 정의하고, 변칙이 감지된 경우 취하는 조치를 지정합니다. 보안 프로파일은 해당 프로파일과 비교하여 평가되는 디바이스를 세부적으로 제어할 수 있도록 가장 구체적인 대상에 연결해야 합니다.

등록되지 않은 디바이스는 디바이스 수명 동안 일관된 MQTT 클라이언트 식별자 또는 사물 이름(디바이스 지표를 보고하는 디바이스의 경우)을 제공하여 모든 위반 및 지표가 동일한 디바이스에 기인하도록 해야 합니다.

> **Important**
>
> 사물 이름에 제어 문자가 포함되어 있거나 사물 이름이 UTF-8로 인코딩된 문자 128바이트보다 긴 경우 디바이스에서 보고된 메시지는 거부됩니다.

보안 사용 사례

이 단원에서는 디바이스 집합을 위협하는 다양한 유형의 공격과 이러한 공격을 모니터링하는 데 사용할 수 있는 권장 지표에 대해 설명합니다. 보안 문제를 조사하기 위한 시작점으로 지표 이상(metric anomaly)을 사용하는 것이 좋지만 지표 이상만을 기준으로 보안 위협을 판단해서는 안 됩니다.

이상 경보를 조사하려면 경보 세부 정보를 디바이스 특성, 디바이스 지표 기록 추세, 보안 프로파일 지표 기록 추세, 사용자 지정 지표 및 로그 등의 다른 컨텍스트 정보와 연관시켜 보안 위협이 있는지 확인합니다.

클라우드 측 사용 사례

Device Defender는 AWS IoT 클라우드 측에서 다음 사용 사례를 모니터링할 수 있습니다.

지적 재산권 도용:

지적 재산권 도용에는 영업 비밀, 하드웨어 또는 소프트웨어를 포함한 개인 또는 회사의 지적 재산권 도용이 포함됩니다. 이것은 종종 디바이스의 제조 단계에서 발생합니다. 지적 재산권 도용은 불법 복제, 디바이스 도용 또는 디바이스 인증서 도용의 형태로 이루어질 수 있습니다. 클라우드 기반 지적 재산권 도용은 IoT 리소스에 의도하지 않은 액세스를 허용하는 정책이 있기 때문에 발생할 수 있습니다. IoT 정책을 검토하고 과도하게 허용적인 검사 감사를 사용하여 지나치게 허용적인 정책을 식별할 수 있습니다.

관련 지표:

지표	이론적 근거
소스 IP	디바이스가 도용된 경우 소스 IP 주소는 정상적인 공급망에서 유통되는 디바이스에 대해 일반적으로 예상되는 IP 주소 범위를 벗어납니다.
수신된 메시지 수	공격자는 클라우드 기반 IP 도용에서 디바이스를 사용할 수 있기 때문에 AWS IoT 클라우드

지표	이론적 근거
메시지 크기	에서 디바이스로 전송된 메시지 수 또는 메시지 크기와 관련된 지표가 급증하여 보안 문제가 발생할 수 있음을 나타냅니다.

MQTT 기반 데이터 유출:

데이터 유출은 악의적인 공격자가 IoT 배포 또는 디바이스에서 무단 데이터 전송을 수행할 때 발생합니다. 공격자는 클라우드 측 데이터 소스에 대해 MQTT를 통해 이러한 유형의 공격을 시작합니다.

관련 지표:

지표	이론적 근거
소스 IP	디바이스가 도용된 경우 소스 IP 주소는 표준 공급망에서 유통되는 디바이스에 대해 일반적으로 예상되는 IP 주소 범위를 벗어납니다.
수신된 메시지 수	공격자는 MQTT 기반 데이터 유출에서 디바이스를 사용할 수 있기 때문에 AWS IoT 클라우드에서 디바이스로 전송된 메시지 수 또는 메시지 크기와 관련된 지표가 급증하여 보안 문제가 발생할 수 있음을 나타냅니다.
메시지 크기	

사칭:

사칭 공격은 공격자가 AWS IoT 클라우드 측 서비스, 애플리케이션, 데이터에 액세스하거나 IoT 디바이스의 명령 및 제어에 참여하기 위해 알려진 엔터티 또는 신뢰할 수 있는 엔터티로 가장하여 공격하는 것입니다.

관련 지표:

지표	이론적 근거
권한 부여 실패	공격자가 도난된 자격 증명을 사용하여 신뢰할 수 있는 엔터티로 가장할 경우 자격 증명이 더

지표	이론적 근거
연결 시도	이상 유효하지 않거나 신뢰할 수 있는 디바이스에서 이미 사용되었을 수 있으므로 연결 관련 지표가 급증하는 경우가 많습니다. 권한 부여 실패, 연결 시도 또는 연결 끊기의 비정상적인 동작은 잠재적인 사칭 시나리오를 가리킵니다.
연결 끊기	

클라우드 인프라 남용:

AWS IoT 클라우드 서비스 남용은 메시지 볼륨이 높거나 큰 크기의 메시지로 주제를 게시하거나 구독할 때 발생합니다. 명령 및 제어에 대한 과도하게 허용된 정책 또는 디바이스 취약성 악용은 클라우드 인프라 남용을 유발할 수도 있습니다. 이 공격의 주요 목표 중 하나는 AWS 청구서를 증가시키는 것입니다. IoT 정책을 검토하고 과도하게 허용적인 검사 감사를 사용하여 지나치게 허용적인 정책을 식별할 수 있습니다.

관련 지표:

지표	이론적 근거
수신된 메시지 수	이 공격의 목적은 메시지 수, 수신된 메시지 및 메시지 크기와 같은 활동을 모니터링하는 지표인 AWS 청구서를 증가시키는 것입니다..
전송된 메시지 수	
메시지 크기	
소스 IP	공격자가 메시징 볼륨을 생성하는 의심스러운 소스 IP 목록이 나타날 수 있습니다.

디바이스 측 사용 사례

Device Defender는 디바이스 측에서 다음 사용 사례를 모니터링할 수 있습니다.

서비스 거부 공격:

DoS(서비스 거부) 공격은 디바이스 또는 네트워크를 종료하여 의도한 사용자가 디바이스 또는 네트워크에 액세스할 수 없도록 하는 것을 목표로 합니다. DoS 공격은 대상에 트래픽이 넘치거나 시

스템 속도가 느려지거나 시스템 실패를 유발하는 요청을 전송하여 액세스를 차단합니다. IoT 디바이스는 DoS 공격에 사용될 수 있습니다.

관련 지표:

지표	이론적 근거
전송된 패킷	DoS 공격은 일반적으로 주어진 디바이스에서 더 높은 아웃바운드 통신율을 포함하며 DoS 공격 유형에 따라 전송된 패킷 및 바이트 수가 모두 증가할 수 있습니다.
전송된 바이트	
목적지 IP	디바이스가 통신해야 하는 IP 주소/CIDR 범위를 정의하면 대상 IP의 이상 현상이 디바이스에서 인증되지 않은 IP 통신을 나타낼 수 있습니다.
수신 TCP 포트	DoS 공격은 일반적으로 디바이스에 설치된 맬웨어가 공격 대상 및 공격 시기에 대한 명령과 정보를 수신하는 더 큰 명령 및 제어 인프라가 필요합니다. 따라서 이러한 정보를 수신하기 위해 맬웨어는 디바이스에서 일반적으로 사용하지 않는 포트에서 수신 대기합니다.
수신 TCP 포트 수	
수신 UDP 포트	
수신 UDP 포트 수	

측면 위협 에스컬레이션:

측면 위협 에스컬레이션은 일반적으로 공격자가 네트워크의 한 지점(예: 연결된 디바이스)에 액세스하는 것으로 시작됩니다. 그런 다음 공격자는 도용된 자격 증명 또는 취약점 악용 등의 방법을 통해 자신의 권한 수준 또는 다른 디바이스에 대한 액세스를 높이려고 시도합니다.

관련 지표:

지표	이론적 근거
전송된 패킷	일반적인 상황에서 공격자는 정찰(reconnaissance)을 수행하고 공격 대상 선택 범위를 좁힐 수 있도록 사용 가능한 디바이스를 식별하기 위해 근거리 통신망에서 스캔을 실행해야
전송된 바이트	

지표	이론적 근거
	합니다. 이러한 종류의 스캔으로 인해 전송된 바이트 및 패킷 수가 급증할 수 있습니다.
목적지 IP	디바이스가 알려진 IP 주소 또는 CIDR 집합과 통신해야 하는 경우 비정상적인 IP 주소와 통신하려고 하는지 식별할 수 있습니다. 이 주소는 측면 위협 에스컬레이션 사용 사례에서 로컬 네트워크의 프라이빗 IP 주소일 수 있습니다.
권한 부여 실패	공격자는 IoT 네트워크에서 권한 수준을 높이려고 할 때 해지되었거나 만료된 도난된 자격 증명을 사용하여 권한 부여 실패를 증가시킬 수 있습니다.

데이터 유출 또는 감시:

데이터 유출은 멀웨어 또는 악의적인 공격자가 디바이스 또는 네트워크 엔드포인트에서 무단 데이터 전송을 수행할 때 발생합니다. 데이터 유출은 일반적으로 공격자가 데이터 또는 지적 재산권을 얻거나 네트워크의 정찰(reconnaissance)을 수행하기 위한 두 가지 목적을 제공합니다. 정찰이란 자격 증명을 도용하고 정보를 수집하기 위해 사용자 활동을 모니터링하는 데 악성 코드가 사용됨을 의미합니다. 아래 지표는 두 유형의 공격을 조사하는 시작 지점을 제공할 수 있습니다.

관련 지표:

지표	이론적 근거
전송된 패킷	데이터 유출이나 감시 공격이 발생하면 공격자는 데이터를 단순히 리디렉션하지 않고 디바이스에서 전송되는 데이터를 미러링하는 경우가 많으며, 이 데이터는 의도된 데이터가 들어오는 것을 보지 못할 때 디펜더에 의해 식별됩니다. 이러한 미러링된 데이터는 디바이스에서 전송되는 총 데이터 양을 크게 증가시켜 전송되는 패킷 및 바이트 수가 급증합니다.
전송된 바이트	

지표	이론적 근거
목적지 IP	공격자가 데이터 유출 또는 감시 공격에 디바이스를 사용하는 경우 공격자가 제어하는 비정상적인 IP 주소로 데이터를 전송해야 합니다. 대상 IP를 모니터링하면 이러한 공격을 식별하는 데 도움이 될 수 있습니다.

암호화폐 채굴

공격자는 암호화폐 채굴을 위해 디바이스의 처리 능력을 활용합니다. 암호화폐 채굴은 컴퓨팅 집약적인 프로세스이며 일반적으로 다른 채굴 피어 및 풀과의 네트워크 통신이 필요합니다.

관련 지표:

지표	이론적 근거
목적지 IP	네트워크 통신은 일반적으로 암호화폐 채굴 중 요구 사항입니다. 디바이스가 통신해야 하는 IP 주소 목록을 엄격하게 제어하면 암호화폐 채굴과 같은 디바이스에서 의도하지 않은 통신을 식별하는 데 도움이 될 수 있습니다.
CPU 사용량 사용자 지정 지표	암호화폐 채굴은 디바이스 CPU 사용량이 높은 컴퓨팅 집약적 작업이 요구됩니다. 이 지표를 수집하고 모니터링하도록 선택하면 정상보다 높은 CPU 사용량이 암호화폐 채굴 작업을 나타내는 지표가 될 수 있습니다.

명령 및 제어, 멀웨어 및 랜섬웨어

맬웨어 또는 랜섬웨어는 디바이스에 대한 제어를 제한하고 디바이스 기능을 제한합니다. 랜섬웨어 공격의 경우 랜섬웨어가 사용하는 암호화로 인해 데이터 액세스가 손실됩니다.

관련 지표:

지표	이론적 근거
목적지 IP	네트워크 또는 원격 공격은 IoT 디바이스에 대한 공격의 상당 부분을 나타냅니다. 디바이스와 통신해야 하는 IP 주소 목록을 엄격하게 제어하면 멀웨어 또는 랜섬웨어 공격으로 인한 비정상적인 대상 IP를 식별할 수 있습니다.
수신 TCP 포트	여러 맬웨어 공격에는 디바이스에서 실행할 명령을 전송하는 명령 및 제어 서버를 시작하는 작업이 포함됩니다. 이러한 유형의 서버는 맬웨어 또는 랜섬웨어 작업에 매우 중요하며 열린 TCP/UDP 포트와 포트 수를 면밀히 모니터링하여 식별할 수 있습니다.
수신 TCP 포트 수	
수신 UDP 포트	
수신 UDP 포트 수	

개념

지표

AWS IoT Device Defender Detect는 지표를 사용하여 디바이스의 변칙적 동작을 감지합니다. AWS IoT Device Defender Detect는 보고된 지표 값을 제공한 예상 값과 비교합니다. 이러한 지표는 클라우드 측 지표와 디바이스 측 지표의 두 소스에서 가져올 수 있습니다. ML Detect는 6개의 클라우드 측 지표와 7개의 디바이스 측 지표를 지원합니다. ML Detect에 대해 지원되는 지표 목록은 지원되는 지표 단원을 참조하세요.

AWS IoT 네트워크의 변칙적 동작은 권한 부여 실패 횟수나 AWS IoT를 통해 디바이스가 전송하거나 수신하는 메시지의 수 또는 크기와 같은 클라우드 측 지표를 통해 감지됩니다.

또한 AWS IoT Device Defender Detect는 디바이스가 수신하는 포트, 전송된 바이트 또는 패킷의 수, 또는 디바이스의 TCP 연결 등 AWS IoT 디바이스가 생성한 지표 데이터를 수집, 집계 및 모니터링할 수 있습니다.

클라우드 측 지표를 통해서만 AWS IoT Device Defender Detect를 사용할 수 있습니다. 디바이스 측 지표를 사용하려면 먼저 AWS IoT가 연결된 디바이스 또는 디바이스 게이트웨이에 AWS IoT SDK를 배포하여 지표를 수집하고 이 지표를 AWS IoT에 전송해야 합니다. 디바이스에서 지표 전송 섹션을 참조하세요.

보안 프로파일

보안 프로필은 계정의 디바이스 그룹(정적 사물 그룹) 또는 모든 디바이스에 대해 이상 동작을 정의하고, 변칙이 감지된 경우 취하는 조치를 지정합니다. AWS IoT 콘솔 또는 API 명령을 사용하여 보안 프로파일을 생성하고 해당 프로파일을 디바이스 그룹에 연결할 수 있습니다. AWS IoT Device Defender Detect는 보안 관련 데이터의 기록을 시작하고, 보안 프로파일에 정의된 동작을 사용하여 디바이스 동작의 변칙을 감지합니다.

동작

동작은 AWS IoT Device Defender Detect가 비정상적인 동작을 보일 때 이를 인식하는 방법을 시스템에 알려줍니다. 동작과 일치하지 않는 모든 디바이스 작업은 알림을 트리거합니다. Rules Detect 동작은 예상된 디바이스 동작을 설명하는 지표와 절대값 또는 연산자(예: 작거나 같음, 크거나 같음)를 사용한 통계 임계값으로 구성됩니다. ML Detect 동작은 지표와 ML Detect 구성으로 이루어지며, 이 구성은 디바이스의 정상적인 동작을 학습하도록 ML 모델을 설정합니다.

ML 모델

ML 모델은 고객이 구성하는 각 동작을 모니터링하기 위해 만들어진 기계 학습 모델입니다. 이 모델은 대상 디바이스 그룹의 지표 데이터 패턴을 학습하고 지표 기반 동작에 대해 세 가지 이상 신뢰(anomaly confidence) 임계값(높음, 중간 및 낮음)을 생성합니다. 디바이스 수준에서 수집된 지표 데이터를 기반으로 이상 현상을 추론합니다. ML Detect의 맥락에서 하나의 ML 모델을 만들어 하나의 지표 기반 동작을 평가합니다. 자세한 내용은 ML Detect 단원을 참조하십시오.

신뢰 수준

ML Detect는 `High`, `Medium`, `Low`의 세 가지 신뢰 수준을 지원합니다. `High` 신뢰도는 비정상적인 동작 평가에서 민감도가 낮고 경보 발생 빈도가 적음을 의미합니다. `Medium` 신뢰도는 중간 민감도를 나타내며, `Low` 신뢰도는 높은 민감도와 경보 발생 빈도가 높음을 의미합니다.

차원

차원을 정의하여 동작 범위를 조정할 수 있습니다. 예를 들어 패턴과 일치하는 MQTT 주제에 동작을 적용하는 주제 필터 차원을 정의할 수 있습니다. 보안 프로파일에서 사용할 차원 정의에 대한 자세한 내용은 차원 생성 단원을 참조하세요.

경보

이상이 감지되면 CloudWatch 지표(https://docs.aws.amazon.com/iot/latest/developerguide/monitoring-cloudwatch.htmlAWS IoT 개발자 안내서의 Amazon CloudWatch를 사용하여 AWS IoT Core 경보 및 지표 모니터링 참조) 또는 SNS 알림을 통해 경보 알림을 전송할 수 있습니다. 경보 알림은 경보에 대한 정보 및 디바이스에 대한 경보 기록과 함께 AWS IoT 콘솔에도 표시됩니다. 모니

터링된 디바이스가 변칙적 동작을 나타내지 않거나 경보가 발생했지만 장기간 보고를 중지한 경우에도 경보가 전송됩니다.

경보 확인 상태

경보가 생성된 후 경보를 참 양성(True positive), 양성(Benign positive), 거짓 양성(False positive) 또는 알 수 없음(Unknown)으로 확인할 수 있습니다. 경보 확인 상태에 설명을 추가할 수도 있습니다. 4가지 확인 상태 중 하나를 사용하여 AWS IoT Device Defender 경보를 보고, 구성하고, 필터링할 수 있습니다. 경보 확인 상태 및 관련 설명을 사용하여 팀원에게 알릴 수 있습니다. 이렇게 하면 팀에서 참 양성 경보에 대한 완화 조치 수행, 양성 경보 건너뛰기, 알 수 없는 경보에 대한 조사 계속 등의 후속 조치를 취할 수 있습니다. 모든 경보의 기본 확인 상태는 알 수 없음(Unknown)입니다.

경보 금지(alarm suppressions)

동작 알림을 on 또는 `suppressed`로 설정하여 Detect 경보 SNS 알림을 관리합니다. 경보를 금지해도 Detect가 디바이스 동작 평가를 수행하는 것을 중지하지 않습니다. Detect는 비정상적인 동작을 위반 경보 플래그로 계속 지정합니다. 그러나 금지된 경보는 SNS 알림에 전달되지 않습니다. 이는 AWS IoT 콘솔 또는 API를 통해 액세스할 수 있습니다.

동작

보안 프로파일에는 여러 가지 동작이 포함되어 있습니다. 각 동작에는 사용자 계정에 속하는 전체 디바이스 또는 디바이스 그룹의 정상 동작을 지정하는 지표가 있습니다. 동작은 Rules Detect 동작과 ML Detect 동작의 두 가지 범주로 나뉩니다. Rules Detect 동작을 통해 디바이스 작동 방식을 정의하는 반면, ML Detect를 통해 기록 디바이스 데이터를 기반으로 구축된 ML 모델을 사용하여 디바이스의 작동 방식을 평가할 수 있습니다.

보안 프로파일은 ML 또는 Rule 기반의 두 가지 임계값 유형 중 하나일 수 있습니다. ML 보안 프로파일은 과거 데이터를 학습하여 디바이스 전반의 디바이스 수준 운영 및 보안 이상을 자동으로 감지합니다. 규칙 기반 보안 프로파일을 사용하려면 디바이스 동작을 모니터링하기 위해 정적 규칙을 수동으로 설정해야 합니다.

아래에서 `behavior` 정의에 사용되는 몇 가지 필드를 설명합니다.

Rules Detect 및 ML Detect에 공통

`name`

동작의 이름입니다.

`metric`

사용된 지표(즉, 동작으로 측정된 항목)의 이름입니다.

`consecutiveDatapointsToAlarm`

디바이스가 지정된 수의 연속적인 데이터 요소에 대한 동작을 위반하면 경보가 발생합니다. 지정하지 않은 경우 기본값은 1입니다.

`consecutiveDatapointsToClear`

경보가 발생한 후 위반 디바이스가 지정된 수의 연속적인 데이터 요소에 대한 동작을 더 이상 위반하지 않으면 경보가 지워집니다. 지정하지 않은 경우 기본값은 1입니다.

`threshold type`

보안 프로파일은 ML 또는 Rules 기반의 두 가지 임계값 유형 중 하나일 수 있습니다. ML 보안 프로파일은 과거 데이터를 학습하여 디바이스 전반의 디바이스 수준 운영 및 보안 이상을 자동으로 감지합니다. 규칙 기반 보안 프로파일을 사용하려면 디바이스 동작을 모니터링하기 위해 정적 규칙을 수동으로 설정해야 합니다.

`alarm suppressions`

동작 알림을 on 또는 `suppressed`로 설정하여 탐지 경보 Amazon SNS 알림을 관리할 수 있습니다. 경보를 금지해도 Detect가 디바이스 동작 평가를 수행하는 것을 중지하지 않습니다. Detect는 비정상적인 동작을 위반 경보 플래그로 계속 지정합니다. 하지만 표시되지 않은 경보는 Amazon SNS 알림에 전달되지 않습니다. 이는 AWS IoT 콘솔 또는 API를 통해서만 액세스할 수 있습니다.

Rules Detect

`dimension`

차원을 정의하여 동작 범위를 조정할 수 있습니다. 예를 들어 패턴과 일치하는 MQTT 주제에 동작을 적용하는 주제 필터 차원을 정의할 수 있습니다. 보안 프로파일에 사용할 차원을 정의하려면 차원 생성을 참조하세요. Rules Detect에만 적용됩니다.

`criteria`

디바이스가 `metric`과 관련하여 정상적으로 동작하는지 여부를 확인하는 기준입니다.

> **Note**
>
> AWS IoT 콘솔에서 알림을 선택하여 AWS IoT Device Defender가 디바이스의 이상 동작을 탐지하면 Amazon SNS를 통해 알림을 받을 수 있습니다.

`comparisonOperator`

측정된 사물(`metric`)을 기준(`value` 또는 `statisticalThreshold`)에 연결하는 연산자입니다.

가능한 값은 "less-than", "less-than-equals", "greater-than", "greater-than-equals", "in-cidr-set", "not-in-cidr-set", "in-port-set", "not-in-port-set"입니다. 모든 지표에 모든 연산자를 사용할 수 있는 것은 아닙니다. CIDR 세트 및 포트용 연산자는 그러한 개체와 관련된 지표에만 사용해야 합니다.

`value`

`metric`과 비교되는 값입니다. 여기에는 지표 유형에 따라 `count`(값), `cidrs`(CIDR 목록) 또는 `ports`(포트 목록)가 포함됩니다.

`statisticalThreshold`

동작 위반을 결정하는 통계 임계값입니다. 이 필드에는 `statistic` 필드가 포함되어 있으며, 가능한 값은 "p0", "p0.1", "p0.01", "p1", "p10", "p50", "p90", "p99", "p99.9", "p99.99" 또는 "p100"입니다.

이 `statistic`은 백분위수를 나타냅니다. 동작에 대한 규정 준수를 결정하는 값으로 해석됩니다. 지표는 이 보안 프로파일과 연결된 모든 보고 디바이스로부터 지정된 기간(`durationSeconds`) 동안 여러 번 수집되며, 백분위수는 이 데이터를 기반으로 계산됩니다. 그 후 디바이스에 대한 측정치가 수집되고 동일한 기간 동안 누적됩니다. 디바이스의 결과 값이 지정된 백분위수와 관련된 값보다 크거나 작으면(`comparisonOperator`), 디바이스가 동작을 준수하고 있다고 간주됩니다. 그렇지 않은 경우 디바이스가 동작을 위반한 것입니다.

백분위수는 관련된 값을 하회한다고 간주되는 모든 측정치의 백분율을 나타냅니다. 예를 들어 "p90"(90번째 백분위수)과 관련된 값이 123인 경우, 모든 측정치의 90%는 123 미만입니다.

`durationSeconds`

시간 차원이 있는 기준에 따라 동작을 평가할 때 그 기간을 지정하려면 사용합니다(예: `NUM_MESSAGES_SENT`). `statisticalThreshhold` 지표 비교의 경우,

`statisticalThreshold` 값을 결정하기 위해 모든 디바이스에 대해 측정치를 수집한 후, 각 디바이스의 동작을 비교하여 순위를 정하는 방법을 결정하기 위해 각 디바이스에 대해 측정치를 수집하는 기간입니다.

ML Detect

`ML Detect confidence`

ML Detect는 `High`, `Medium`, `Low`의 세 가지 신뢰 수준을 지원합니다. `High` 신뢰도는 비정상적인 동작 평가에서 민감도가 낮고 경보 발생 빈도가 적음을 의미합니다. `Medium` 신뢰도는 중간 민감도를 나타내며, `Low` 신뢰도는 높은 민감도와 경보 발생 빈도가 높음을 의미합니다.

ML Detect

기계 학습 Detect(ML Detect)를 사용하면 기계 학습을 사용하여 기록 디바이스 데이터를 기반으로 모델을 자동으로 생성함으로써 예상되는 디바이스 동작을 학습하고 이러한 프로파일을 디바이스 그룹 또는 집합의 모든 디바이스에 할당하는 보안 프로파일을 생성할 수 있습니다. AWS IoT Device Defender에서는 이상 현상을 식별하고 ML 모델을 사용하여 경보를 트리거합니다.

ML Detect를 시작하는 방법에 대한 자세한 내용은 ML Detect 가이드 단원을 참조하세요.

이번 장은 다음과 같은 단원들로 구성되어 있습니다.

- ML Detect의 사용 사례
- ML Detect 작동 방식
- 최소 요구 사항
- 제한 사항
- 경보에서 거짓 양성 및 기타 확인 상태 표시
- 지원되는 지표
- 서비스 할당량
- ML Detect CLI 명령
- ML Detect API
- ML Detect 보안 프로파일 일시 중지 또는 삭제

ML Detect의 사용 사례

ML Detect를 사용하여 디바이스의 예상 동작을 설정하기가 어려울 때 플릿 디바이스를 모니터링할 수 있습니다. 예를 들어 연결 해제 지표 수를 모니터링하기 위해 허용되는 임계값으로 간주되는 것이 명확하지 않을 수 있습니다. 이 경우 ML Detect를 활성화하여 디바이스에서 보고된 기록 데이터를 기반으로 비정상적인 연결 해제 지표 데이터 포인트를 식별할 수 있습니다.

ML Detect의 또 다른 사용 사례는 시간이 지남에 따라 동적으로 변하는 디바이스 동작을 모니터링하는 것입니다. ML Detect는 디바이스의 데이터 패턴 변경을 기반으로 예상되는 동적 디바이스 동작을 주기적으로 학습합니다. 예를 들어, 디바이스 메시지 전송 볼륨은 평일과 주말에 따라 다를 수 있으며 ML Detect는 이러한 동적 동작을 학습합니다.

ML Detect 작동 방식

ML Detect를 사용하면 6가지 클라우드 측 지표 및 7개의 디바이스 측 지표 간에 운영 및 보안 이상을 식별하는 동작을 생성할 수 있습니다. 초기 모델 훈련 기간이 끝나면 ML Detect는 14일의 데이터를 기준으로 매일 모델을 새로 고칩니다. ML 모델을 사용하여 이러한 지표에 대한 데이터 포인트를 모니터링하고 예외 항목이 감지되면 경보를 트리거합니다.

ML Detect는 유사한 예상 동작을 가진 디바이스 컬렉션에 보안 프로파일을 연결하는 경우에 가장 적합합니다. 예를 들어 일부 디바이스가 고객의 가정과 사무실의 다른 디바이스에서 사용되는 경우 디바이스 동작 패턴이 두 그룹 간에 크게 다를 수 있습니다. 디바이스를 가정용 디바이스(home-device) 사물 그룹과 사무실 디바이스(office-device) 사물 그룹으로 구성할 수 있습니다. 최상의 이상 감지 효과를 얻으려면 각 사물 그룹을 별도의 ML Detect 보안 프로파일에 연결합니다.

ML Detect가 초기 모델을 빌드하는 동안 모델을 생성하려면 후행 14일 동안 지표당 최소 25,000개의 데이터 포인트가 필요합니다. 그런 다음 최소 수의 지표 데이터 포인트가 매일 모델을 업데이트합니다. 최소 요구 사항이 충족되지 않으면 ML Detect는 다음 날 모델을 만들려고 시도하며 모델 평가를 중단하기 전의 다음 30일 동안 매일 다시 시도합니다.

최소 요구 사항

초기 ML 모델 교육 및 생성을 위해 ML Detect에는 다음과 같은 최소 요구 사항이 있습니다.

최소 교육 기간

초기 모델을 제작하는 데 14일이 걸립니다. 그런 다음 모델은 14일의 후행 기간의 지표 데이터로 매일 새로 고쳐집니다.

최소 총 데이터 포인트

ML 모델을 빌드하는 데 필요한 최소 데이터 포인트는 지난 14일 동안 지표당 25,000개의 데이터 포인트입니다. 모델을 지속적으로 교육하고 새로 고치려면 ML Detect를 모니터링하는 디바이스에서 최소 데이터 포인트를 충족해야 합니다. 대략 다음 설정과 동일합니다.

- 45분의 간격으로 AWS IoT에서 연결 및 활동을 하는 디바이스 60개.
- 30분의 간격으로 디바이스 40개
- 10분의 간격으로 디바이스 15개.
- 5분의 간격으로 디바이스 7개.

디바이스 그룹 대상

데이터를 수집하려면 보안 프로파일의 대상 사물 그룹에 사물이 있어야 합니다.

초기 모델이 생성된 후 ML 모델은 매일 새로 고쳐지고 14일 후행 기간 동안 최소 25,000개의 데이터 포인트가 필요합니다.

제한 사항

다음 클라우드 측 지표의 차원과 함께 ML Detect를 사용할 수 있습니다.

- 권한 부여 실패(aws:num-authorization-failures)
- 수신된 메시지(aws:num-messages-received)
- 전송된 메시지(aws:num-messages-sent)
- 메시지 크기(aws:message-byte-size)

다음 지표는 ML Detect에서 지원되지 않습니다.

ML Detect에서 지원되지 않는 클라우드 측 지표:

- 소스 IP(aws:source-ip-address)

ML Detect에서 지원되지 않는 디바이스 측 지표:

- 대상 IP(`aws:destination-ip-addresses`)
- 수신 TCP 포트(`aws:listening-tcp-ports`)

- 수신 UDP 포트(aws:listening-udp-ports)

사용자 지정 지표는 번호유형만 지원합니다.

경보에서 거짓 양성 및 기타 확인 상태 표시

조사를 통해 ML Detect 경보가 거짓 긍정임을 확인하면 경보의 확인 상태를 거짓 긍정(False positive)으로 설정할 수 있습니다. 이를 통해 사용자와 팀이 응답할 필요가 없는 경보를 식별할 수 있습니다. 경보를 참 양성(True positive), 양성(Benign positive) 또는 알 수 없음(Unknown)으로 표시할 수도 있습니다.

AWS IoT Device Defender 콘솔을 통해 또는 PutVerificationStateOnViolation API 작업을 사용하여 경보를 표시할 수 있습니다.

지원되는 지표

ML Detect와 함께 다음과 같은 클라우드 측 지표를 사용할 수 있습니다.

- 권한 부여 실패(aws:num-authorization-failures)
- 연결 시도(aws:num-connection-attempts)
- 연결 해제(aws:num-disconnects)
- 메시지 크기(aws:message-byte-size)
- 전송된 메시지(aws:num-messages-sent)
- 수신된 메시지(aws:num-messages-received)

ML Detect와 함께 다음과 같은 디바이스 측 지표를 사용할 수 있습니다.

- 전송된 바이트(aws:all-bytes-out)
- 수신된 바이트(aws:all-bytes-in)
- 수신 TCP 포트 개수(aws:num-listening-tcp-ports)
- 수신 UDP 포트 수aws:num-listening-udp-ports)
- 전송된 패킷(aws:all-packets-out)
- 수신된 패킷(aws:all-packets-in)
- 설정된 TCP 연결 수(aws:num-established-tcp-connections)

서비스 할당량

ML Detect 서비스 할당량 및 제한에 대한 자세한 내용은 AWS IoT Device Defender 엔드포인트 및 할당량을 참조하세요.

ML Detect CLI 명령

다음 CLI 명령을 사용하여 ML Detect를 만들고 관리할 수 있습니다.

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

ML Detect API

다음 API를 사용하여 ML Detect 보안 프로파일을 만들고 관리할 수 있습니다.

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

ML Detect 보안 프로파일 일시 중지 또는 삭제

ML Detect 보안 프로파일을 일시 중지하여 디바이스 동작 모니터링을 일시적으로 중지하거나 ML Detect 보안 프로파일을 삭제하여 장시간 디바이스 동작 모니터링을 중지할 수 있습니다.

콘솔을 사용하여 ML Detect 보안 프로필 일시 중지

콘솔을 사용하여 ML Detect 보안 프로파일을 일시 중지하려면 먼저 빈 사물 그룹이 있어야 합니다. 빈 사물 그룹을 만들려면 AWS IoT Core 개발자 안내서의 정적 사물 그룹을 참조하세요. 빈 사물 그룹을 생성한 경우 빈 사물 그룹을 ML Detect 보안 프로파일의 대상으로 설정합니다.

> **Note**
>
> 보안 프로파일의 대상을 30일 이내에 디바이스가 있는 디바이스 그룹으로 다시 설정해야 합니다. 그렇지 않으면 보안 프로파일을 다시 활성화할 수 없습니다.

콘솔을 사용하여 ML Detect 보안 프로필 삭제

보안 프로파일을 삭제하려면 다음 단계를 따릅니다.

1. AWS IoT 콘솔에서 사이드바로 이동하여 방어 섹션을 선택합니다.
2. 방어에서 감지를 선택하고 나서 보안 프로파일을 선택합니다.
3. 삭제하려는 ML 보안 프로파일을 선택합니다.
4. 작업(Actions)을 선택하고 옵션에서 삭제(Delete)를 선택합니다.

> **Note**
>
> ML Detect 보안 프로파일을 삭제한 후에는 보안 프롤파일을 다시 활성화할 수 없습니다.

CLI를 사용하여 ML Detect 보안 프로필 일시 중지

CLI를 사용하여 ML Detect 보안 프로파일을 일시 중지하려면 `detach-security-security-profile` 명령 사용

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

> **Note**
>
> 이 옵션은 AWS CLI에서만 사용할 수 있습니다. 콘솔 워크플로와 마찬가지로 보안 프로파일의 대상을 30일 이내에 디바이스가 있는 디바이스 그룹으로 다시 설정해야 합니다. 그렇지 않으면 보안 프로파일을 다시 활성화할 수 없습니다. 보안 프로파일을 디바이스 그룹에 연결하려면 `attach-security-profile` 명령을 사용하세요.

CLI를 사용하여 ML Detect 보안 프로필 삭제

아래 `delete-security-profile` 명령을 사용하여 보안 프로파일을 삭제할 수 있습니다.

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

> **Note**
>
> ML Detect 보안 프로파일을 삭제한 후에는 보안 프로파일을 다시 활성화할 수 없습니다.

사용자 지정 지표

AWS IoT Device Defender 사용자 지정 지표를 사용하면 Wi-Fi 게이트웨이에 연결된 디바이스 수, 배터리 충전 수준 또는 스마트 플러그의 전원 사이클 수 등 플릿 또는 사용 사례에 고유한 지표를 정의하고 모니터링할 수 있습니다. 사용자 지정 지표 동작은 디바이스 그룹(사물 그룹) 또는 모든 디바이스에 대해 예상되는 동작을 지정하는 보안 프로파일에 정의됩니다. 경보를 설정하여 동작을 모니터링할 수 있습니다. 이 경보를 사용하면 디바이스와 관련된 문제를 감지하고 이에 대응할 수 있습니다.

이번 장은 다음과 같은 단원들로 구성되어 있습니다.

- 콘솔에서 사용자 지정 지표를 사용하는 방법
- CLI에서 사용자 지정 지표를 사용하는 방법
- 사용자 지정 지표 CLI 명령
- 사용자 지정 지표 API

콘솔에서 사용자 지정 지표를 사용하는 방법

튜토리얼

- AWS IoT Device Defender 에이전트 SDK(Python)
- 사용자 지정 지표를 생성하여 보안 프로파일에 추가
- 사용자 지정 지표 세부 정보 보기
- 사용자 지정 지표 업데이트
- 사용자 지정 지표 삭제

AWS IoT Device Defender 에이전트 SDK(Python)

시작하려면 AWS IoT Device Defender Agent SDK(Python) 샘플 에이전트를 다운로드하세요. 에이전트는 지표를 수집하고 보고서를 게시합니다. 디바이스 측 지표가 게시되면 수집 중인 지표를 보고 경보 설정을 위한 임계값을 확인할 수 있습니다. 디바이스 에이전트 설정에 대한 지침은 AWS IoT Device Defender 에이전트 SDK(Python) Readme에서 확인할 수 있습니다. 자세한 내용은 AWS IoT Device Defender 에이전트 SDK(Python) 단원을 참조하세요.

사용자 지정 지표를 생성하여 보안 프로파일에 추가

다음 절차는 콘솔에서 사용자 지정 지표를 생성하는 방법을 보여줍니다.

1. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 지표를 차례로 선택합니다.
2. 사용자 지정 지표 페이지에서 생성을 선택합니다.
3. 사용자 지정 지표 생성 페이지에서 다음 작업을 수행합니다.

   1. 이름에서 사용자 지정 지표의 이름을 입력합니다. 사용자 지정 지표를 생성한 후에는 이 이름을 수정할 수 없습니다.
   2. 표시 이름(선택 사항)에서 사용자 지정 지표의 친근한 이름을 입력할 수 있습니다. 고유할 필요는 없으며 생성 후 수정할 수 있습니다.
   3. 유형에서 모니터링할 지표의 유형을 선택합니다. 지표 유형은 string-list, ip-address-list, number-list 및 number를 포함합니다. 유형을 생성한 후에는 수정할 수 없습니다.

      > **Note**
      >
      > ML Detect는 숫자 유형만 허용합니다.

   4. 태그에서 리소스와 연결할 태그를 선택할 수 있습니다.

   마쳤으면 확인(Confirm)을 선택합니다.

4. 사용자 지정 지표를 만든 후 사용자 지정 지표 페이지가 나타나면 새로 만든 사용자 지정 지표를 볼 수 있습니다.
5. 그런 다음 사용자 지정 지표를 보안 프로파일에 추가해야 합니다. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 보안 프로파일을 차례로 선택합니다.
6. 사용자 지정 지표를 추가할 보안 프로파일을 선택합니다.
7. 작업(Actions), 편집(Edit)을 선택합니다.
8. 보존할 추가 지표(Additional Metrics to retain)를 선택한 다음 사용자 지정 지표를 선택합니다. 확인(Confirm) 페이지가 나타날 때까지 다음 화면에서 다음(Next)을 선택합니다. 저장(Save) 및 계속(Continue)을 선택합니다. 사용자 지정 지표가 성공적으로 추가되면 보안 프로파일 세부 정보 페이지가 나타납니다.

> **Note**
>
> 지표 값에 음수 값이 포함된 지표에서는 백분위수 통계를 사용할 수 없습니다.

사용자 지정 지표 세부 정보 보기

다음 절차는 콘솔에서 사용자 지정 지표의 세부 정보를 보는 방법을 보여줍니다.

1. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 지표를 차례로 선택합니다.
2. 세부 정보를 보려는 사용자 지정 지표의 지표 이름을 선택합니다.

사용자 지정 지표 업데이트

다음 절차는 콘솔에서 사용자 지정 지표를 업데이트하는 방법을 보여줍니다.

1. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 지표를 차례로 선택합니다.
2. 업데이트하려는 사용자 지정 지표 옆에 있는 옵션 버튼을 선택합니다. 그리고 나서 작업에 편집을 선택합니다.
3. 사용자 지정 지표 업데이트 페이지에서 표시 이름을 편집하고 태그를 제거하거나 추가할 수 있습니다.
4. 완료되면 업데이트를 선택합니다. 사용자 지정 지표 페이지로 이동합니다.

사용자 지정 지표 삭제

다음 절차는 콘솔에서 사용자 지정 지표를 삭제하는 방법을 보여줍니다.

1. 먼저 참조된 보안 프로파일에서 사용자 지정 지표를 제거합니다. 사용자 지정 지표 세부 정보 페이지에서 사용자 지정 지표가 포함된 보안 프로파일을 볼 수 있습니다. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 지표를 차례로 선택합니다.
2. 제거할 사용자 지정 지표를 선택합니다. 사용자 지정 지표 세부 정보 페이지에서 보안 프로파일 아래에 나열된 보안 프로파일에서 사용자 지정 지표를 제거합니다.
3. AWS IoT 콘솔의 탐색 창에서 방어를 확장한 다음 감지, 지표를 차례로 선택합니다.
4. 삭제하려는 사용자 지정 지표 옆에 있는 옵션 버튼을 선택합니다. 그리고 나서 작업에 삭제를 선택합니다.
5. 사용자 지정 지표를 삭제하시겠습니까? 메시지에서 사용자 지정 지표 삭제를 선택합니다.

> **Warning**
>
> 사용자 지정 지표를 삭제하면 지표와 연결된 모든 데이터가 손실됩니다. 이 작업은 실행 취소할 수 없습니다.

CLI에서 사용자 지정 지표를 사용하는 방법

튜토리얼

- AWS IoT Device Defender 에이전트 SDK(Python)
- 사용자 지정 지표를 생성하여 보안 프로파일에 추가
- 사용자 지정 지표 세부 정보 보기
- 사용자 지정 지표 업데이트
- 사용자 지정 지표 삭제

AWS IoT Device Defender 에이전트 SDK(Python)

시작하려면 AWS IoT Device Defender Agent SDK(Python) 샘플 에이전트를 다운로드하세요. 에이전트는 지표를 수집하고 보고서를 게시합니다. 디바이스 측 지표를 게시한 후에는 수집 중인 지표를 확인하고 경보 설정을 위한 임계값을 결정할 수 있습니다. 디바이스 에이전트 설정에 대한 지침은 AWS IoT Device Defender 에이전트 SDK(Python) Readme에서 확인할 수 있습니다. 자세한 내용은 AWS IoT Device Defender 에이전트 SDK(Python) 단원을 참조하세요.

사용자 지정 지표를 생성하여 보안 프로파일에 추가

다음 절차에서는 CLI에서 사용자 지정 지표를 생성하여 보안 프로파일에 추가하는 방법을 소개합니다.

1. create-custom-metric 명령을 사용하여 사용자 지정 지표를 만들 수 있습니다. 다음 예에서는 배터리 비율을 측정하는 사용자 지정 지표를 생성합니다.

```
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" \
```

출력:

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

2. 사용자 지정 지표를 만든 후 update-security-profile을(를) 사용하여 기존 프로파일에 사용자 지정 지표를 추가하거나 create-security-profile을(를) 사용하여 사용자 지정 지표를 추가하는 새 보안 프로파일을 생성할 수 있습니다. 여기서는 새 *batteryPercentage* 사용자 지정 지표를 추가하기 위해 *batteryUsage*라는 새 보안 프로파일을 생성합니다. 또한 *cellularBandwidth*라는 Rules Detect 지표를 추가합니다.

```
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
```

출력:

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

> **Note**
> 지표 값에 음수 값이 포함된 지표에서는 백분위수 통계를 사용할 수 없습니다.

사용자 지정 지표 세부 정보 보기

다음 절차에서는 CLI에서 사용자 지정 지표에 대한 세부 정보를 보는 방법을 보여줍니다.

- list-custom-metrics 명령을 사용하여 모든 사용자 지정 지표를볼 수 있습니다.

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

  이 명령의 출력은 다음과 같습니다.

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

사용자 지정 지표 업데이트

다음 절차에서는 CLI에서 사용자 지정 지표를 업데이트하는 방법을 보여줍니다.

- update-custom-metric 명령을 사용하여 사용자 지정 지표를 업데이트합니다. 다음 예제에서는 `display-name`을(를) 업데이트합니다.

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

```
    --region us-east-1
```

이 명령의 출력은 다음과 같습니다.

```
{
    "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"
}
```

사용자 지정 지표 삭제

다음 절차에서는 CLI에서 사용자 지정 지표를 삭제하는 방법을 소개합니다.

1. 사용자 지정 지표를 삭제하려면 먼저 연결되어 있는 보안 프로파일에서 분리합니다. `list-security-profiles` 명령을 사용하여 특정 사용자 지정 지표가 있는 보안 프로파일을 봅니다.
2. 보안 프로파일에서 사용자 지정 지표를 제거하려면 `update-security-profiles` 명령을 사용합니다. 사용자 지정 지표를 제외하고, 유지하려는 모든 정보를 입력합니다.

```
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}}]"
```

이 명령의 출력은 다음과 같습니다.

```
{
  "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. 사용자 지정 지표를 분리한 후에는 delete-custom-metric 명령을 사용하여 사용자 지정 지표를 삭제합니다.

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

이 명령의 출력은 다음과 같습니다.

```
HTTP 200
```

사용자 지정 지표 CLI 명령

다음 CLI 명령을 사용하여 사용자 지정 지표를 만들고 관리할 수 있습니다.

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

사용자 지정 지표 API

다음 CLI 명령을 사용하여 사용자 지정 지표를 만들고 관리할 수 있습니다.

- CreateCustomMetric
- DescribeCustomMetric
- ListCustomMetrics
- UpdateCustomMetric

- DeleteCustomMetric
- ListSecurityProfiles

Device-side metrics

보안 프로파일을 만들 때 IoT 디바이스에서 생성된 지표에 대한 동작 및 임계값을 구성하여 IoT 디바이스의 예상 동작을 지정할 수 있습니다. 다음은 디바이스에 설치한 에이전트의 지표인 디바이스 측 지표입니다.

전송된 바이트(`aws:all-bytes-out`)

지정된 기간 동안 디바이스에서 아웃바운드된 바이트 수입니다.

지정된 기간에 디바이스에서 전송해야 하는 아웃바운드 트래픽의 최대 또는 최소 양(바이트 단위로 측정)을 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 바이트

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

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

```
  "suppressAlerts": true
}
```

Example **statisticalThreshold**를 사용하는 예제

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

Example ML Detect를 사용하는 예

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

수신된 바이트(**aws:all-bytes-in**)

지정된 기간 동안 디바이스로 인바운드된 바이트 수입니다.

지정된 기간에 디바이스에서 수신해야 하는 인바운드 트래픽의 최대 또는 최소 양(바이트 단위로 측정)을 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 바이트

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

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

Example **`statisticalThreshold`**를 사용하는 예제

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

Example ML Detect를 사용하는 예

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

수신 TCP 포트 개수(`aws:num-listening-tcp-ports`)

디바이스가 수신하는 TCP 포트의 수입니다.

이 지표를 사용하여 각 디바이스에서 모니터링해야 하는 TCP 포트의 최대 수를 지정합니다.

호환 가능: Rules Detect | ML Detect

단위: 실패

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 실패

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

```
{
  "name": "Max TCP Ports",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
```

```
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **statisticalThreshold**를 사용하는 예제

```
{
  "name": "Max TCP Ports",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect를 사용하는 예

```
{
  "name": "Max TCP Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

수신 UDP 포트 수**aws:num-listening-udp-ports**)

디바이스가 수신하는 UPD 포트 수입니다.

이 지표를 사용하여 각 디바이스에서 모니터링해야 하는 UDP 포트의 최대 수를 지정합니다.

호환 가능: Rules Detect | ML Detect

단위: 실패

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 실패

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

```
{
  "name": "Max UDP Ports",
  "metric": "aws:num-listening-udp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`**를 사용하는 예제

```
{
  "name": "Max UDP Ports",
  "metric": "aws:num-listening-udp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
```

```
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect를 사용하는 예

```
{
  "name": "Max UPD Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

전송된 패킷(`aws:all-packets-out`)

지정된 기간 동안 디바이스에서 아웃바운드된 패킷 수입니다.

지정된 기간에 디바이스에서 전송해야 하는 전체 아웃바운드 트래픽의 최소 및 최대 양을 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 패킷

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

```
{
  "name": "TCP outbound traffic",
```

```
  "metric": "aws:all-packets-out",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 100
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **statisticalThreshold**를 사용하는 예제

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

Example ML Detect를 사용하는 예

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

```
}
```

수신된 패킷(`aws:all-packets-in`)

지정된 기간 동안 디바이스로 인바운드된 패킷 수입니다.

지정된 기간에 디바이스에서 수신해야 하는 전체 인바운드 트래픽의 최소 및 최대 양을 지정하려면 이 지표를 사용합니다.

호환 가능: Rule Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 패킷

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-packets-in",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 100
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example

`statisticalThreshold`를 사용하는 예제

```
{
```

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

Example ML Detect를 사용하는 예

```
{
  "name": "Inbound sent ML behavior",
  "metric": "aws:all-packets-in",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

대상 IP(`aws:destination-ip-addresses`)

IP 대상 집합입니다.

각 디바이스에서 AWS IoT에 연결해야 하거나 연결하지 않아야 하는 허용된 CIDR 집합(이전 명칭은 화이트리스트) 또는 거부된 Classless Inter-Domain Routings(CIDR) 집합(이전 명칭은 블랙리스트)을 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect

연산자: in-cidr-set | not-in-cidr-set

값: CIDR 목록

단위: 해당 사항 없음

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
}
```

수신 TCP 포트(`aws:listening-tcp-ports`)

디바이스가 수신하는 TCP 포트입니다.

각 디바이스에서 수신하거나 수신하지 않아야 하는 허용된 TCP 포트 집합(이전 명칭은 화이트리스트) 또는 거부된 TCP 포트 집합(이전 명칭은 블랙리스트)을 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect

연산자: in-port-set | not-in-port-set

값: 포트 목록

단위: 해당 사항 없음

Example

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 443, 80 ]
    }
  },
```

```
  "suppressAlerts": true
}
```

수신 UDP 포트(`aws:listening-udp-ports`)

디바이스가 수신하는 UDP 포트입니다.

각 디바이스에서 수신하거나 수신하지 않아야 하는 허용된 UDP 포트 집합(이전 명칭은 화이트리스트) 또는 거부된 UDP 포트 집합(이전 명칭은 블랙리스트)을 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect

연산자: in-port-set | not-in-port-set

값: 포트 목록

단위: 해당 사항 없음

Example

```
{
  "name": "Listening UDP Ports",
  "metric": "aws:listening-udp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 1025, 2000 ]
    }
  }
}
```

설정된 TCP 연결 수(`aws:num-established-tcp-connections`)

디바이스에 대한 TCP 연결 수입니다.

각 디바이스에 있어야 하는 활성 TCP 연결(모든 TCP 상태)의 최대 또는 최소 수를 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 연결

Example

```
{
  "name": "TCP Connection Count",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 3
    },
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`**를 사용하는 예제

```
{
  "name": "TCP Connection Count",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 900,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect를 사용하는 예

```
{
  "name": "Connection count ML behavior",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
```

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

디바이스 지표 문서 사양

전체 구조

긴 이름	짧은 이름	필수	유형	제약 조건	참고
header	hed	Y	객체		올바른 형식의 보고서에 필요한 전체 블록입니다.
지표	met	Y	객체		보고서에는 둘 다 또는 하나 이상의 `metrics` 또는 `custom_metrics` 블록이 있어야 합니다.
custom_metrics	cmet	Y	객체		보고서에는 둘 다 또는 하나 이상의 `metrics` 또는 `custom_metrics` 블록이 있어야 합니다.

헤더 블록

긴 이름	짧은 이름	필수	유형	제약 조건	참고
report_id	rid	Y	Integer		단순 증가 값. Epoch 타임스탬프 권장.
version	v	Y	String	Major.Minor	필드 추가와 함께 작게 증분. 지표가 제거된 경우 크게 증분.

지표 블록:

TCP 연결

긴 이름	짧은 이름	상위 요소	필수	유형	제약 조건	참고
tcp_connections	tc	지표	N	객체		
established_connections	ec	tcp_connections	N	객체		설정된 TCP 상태
연결	cs	established_connections	N	목록<객체>		
remote_addr	rad	연결	Y	숫자	ip:포트	IP는 IPv6 또는 IPv4 일 수 있음
local_port	lp	연결	N	숫자	>= 0	
local_interface	li	연결	N	String		인터페이스 이름

긴 이름	짧은 이름	상위 요소	필수	유형	제약 조건	참고
총합	t	established_connections	N	숫자	>= 0	설정된 연결 수

수신 TCP 포트

긴 이름	짧은 이름	상위 요소	필수	유형	제약 조건	참고
listening_tcp_ports	tp	지표	N	객체		
ports	pts	listening_tcp_ports	N	목록<객체>	> 0	
포트	pt	ports	N	숫자	> 0	포트는 0보다 큰 숫자여야 함
인터페이스	다음과 같은 경우	ports	N	String		인터페이스 이름
총합	t	listening_tcp_ports	N	숫자	>= 0	

수신 UDP 포트

긴 이름	짧은 이름	상위 요소	필수	유형	제약 조건	참고
listening_udp_ports	up	지표	N	객체		
ports	pts	listening_udp_ports	N	목록<포트>	> 0	

긴 이름	짧은 이름	상위 요소	필수	유형	제약 조건	참고
포트	pt	ports	N	숫자	> 0	포트는 0보다 큰 숫자여야 함
인터페이스	다음과 같은 경우	ports	N	String		인터페이스 이름
총합	t	listening_udp_ports	N	숫자	>= 0	

네트워크 통계

긴 이름	짧은 이름	상위 요소	필수	유형	제약 조건	참고
network_stats	ns	지표	N	객체		
bytes_in	bi	network_stats	N	숫자	델타 지표, >= 0	
bytes_out	bo	network_stats	N	숫자	델타 지표, >= 0	
packets_in	pi	network_stats	N	숫자	델타 지표, >= 0	
packets_out	po	network_stats	N	숫자	델타 지표, >= 0	

Example

다음 JSON 구조는 긴 이름을 사용합니다.

```
{
  "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 짧은 이름을 사용하는 예제 JSON 구조

```
{
  "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"
        ]
      }
    ]
  }
}
```

디바이스에서 지표 전송

AWS IoT Device Defender Detect는 AWS IoT 디바이스에서 생성된 지표 데이터를 수집, 집계 및 모니터링하여 비정상적인 동작을 나타내는 디바이스를 식별합니다. 이 섹션에서는 디바이스에서 AWS IoT Device Defender로 지표를 전송하는 방법을 보여줍니다.

디바이스 측 지표를 수집하려면 AWS IoT 커넥티드 디바이스 또는 디바이스 게이트웨이에 AWS IoT SDK 버전 2를 안전하게 배포해야 합니다. 여기에서 SDK의 전체 목록을 확인하세요.

AWS IoT Device Client는 AWS IoT Device Defender 및 AWS IoT Device Management에 있는 기능을 모두 다루는 단일 에이전트를 제공하므로 지표를 게시하는 데 사용할 수 있습니다. 이러한 기능에는 작업, 보안 터널링, AWS IoT Device Defender 지표 게시 등의 작업이 포함됩니다.

AWS IoT Device Defender가 수집 및 평가할 수 있도록 AWS IoT의 예약된 주제(reserved topic)에 디바이스 측 지표를 게시합니다.

지표 게시를 위해 AWS IoT 디바이스 클라이언트 사용

AWS IoT 디바이스 클라이언트를 설치하려면 GitHub에서 다운로드할 수 있습니다. 디바이스 측 데이터를 수집할 디바이스에 AWS IoT 디바이스 클라이언트를 설치한 후 디바이스 측 지표를 AWS IoT Device Defender에 전송하도록 구성해야 합니다. AWS IoT 디바이스 클라이언트 구성 파일의 `device-defender` 섹션에 다음 매개변수가 설정되어 있는지 확인합니다.

```
  "device-defender":    {
        "enabled":    true,
        "interval-in-seconds": 300
    }
```

> **Warning**
> 시간 간격을 최소 300초로 설정해야 합니다. 시간 간격을 300초 미만으로 설정하면 지표 데이터가 제한될 수 있습니다.

구성을 업데이트한 후 AWS IoT Device Defender 콘솔에서 보안 프로파일 및 동작을 생성하여 디바이스가 클라우드에 게시하는 지표를 모니터링할 수 있습니다. 방어(Defend), 감지(Detect), 지표(Metrics)를 차례로 선택하여 AWS IoT Core 콘솔에서 게시된 지표를 찾을 수 있습니다.

클라우드 측 지표

보안 프로파일을 만들 때 IoT 디바이스에서 생성된 지표에 대한 동작 및 임계값을 구성하여 IoT 디바이스의 예상 동작을 지정할 수 있습니다. 다음은 클라우드 측 지표로, AWS IoT의 지표입니다.

메시지 크기(aws:message-byte-size)

메시지의 바이트 수입니다. 디바이스에서 AWS IoT로 전송된 각 메시지의 최대 또는 최소 크기(바이트)를 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 바이트

Example

```
{
  "name": "Max Message Size",
  "metric": "aws:message-byte-size",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 1024
    },
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
```

```
  "suppressAlerts": true
}
```

Example **statisticalThreshold**를 사용하는 예제

```
{

  "name": "Large Message Size",
  "metric": "aws:message-byte-size",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect를 사용하는 예

```
{
  "name": "Message size ML behavior",
  "metric": "aws:message-byte-size",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

3회 연속 5분간 누적 크기가 이 보안 프로파일 동작에 대해 보고하는 다른 모든 디바이스의 90%에 대해 측정된 것보다 큰 메시지를 전송하면 디바이스에 대해 경보가 발생합니다.

전송된 메시지(aws:num-messages-sent)

지정된 기간 동안 디바이스에서 전송한 메시지의 수입니다.

지정된 기간에 AWS IoT와 각 디바이스 간 전송될 수 있는 메시지의 최대 또는 최소 수를 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 메시지

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

```
{

  "name": "Out bound message count",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 50
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
    },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`**를 사용하는 예제

```
{

  "name": "Out bound message rate",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
```

```
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect를 사용하는 예

```
{
  "name": "Messages sent ML behavior",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

수신된 메시지(aws:num-messages-received)

지정된 기간 동안 디바이스에서 수신한 메시지의 수입니다.

지정된 기간에 AWS IoT와 각 디바이스 간 수신될 수 있는 메시지의 최대 또는 최소 수를 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 메시지

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

```
{
```

```
  "name": "In bound message count",
  "metric": "aws:num-messages-received",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 50
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
    },
  "suppressAlerts": true
}
```

Example **statisticalThreshold**를 사용하는 예제

```
{
  "name": "In bound message rate",
  "metric": "aws:num-messages-received",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect를 사용하는 예

```
{
  "name": "Messages received ML behavior",
  "metric": "aws:num-messages-received",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
```

```
  "suppressAlerts": true
}
```

권한 부여 실패(aws:num-authorization-failures)

지정된 기간에 각 디바이스에 허용된 권한 부여 실패의 최대 수를 지정하려면 이 지표를 사용합니다. 디바이스에서 충분한 권한이 없는 주제에 게시하려고 시도하는 경우와 같이 디바이스에서 AWS IoT로의 요청이 거부되는 경우 권한 부여 실패가 발생합니다.

호환 가능: Rules Detect | ML Detect

단위: 실패

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

```
{
  "name": "Authorization Failures",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "comparisonOperator": "less-than",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`**를 사용하는 예제

```
{
  "name": "Authorization Failures",
  "metric": "aws:num-authorization-failures",
```

```
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect를 사용하는 예

```
{
  "name": "Authorization failures ML behavior",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

소스 IP(aws:source-ip-address)

디바이스가 AWS IoT에 연결된 IP 주소입니다.

각 디바이스에서 AWS IoT에 연결해야 하거나 연결하지 않아야 하는 허용된 CIDR 집합(이전 명칭은 화이트리스트) 또는 거부된 Classless Inter-Domain Routings(CIDR) 집합(이전 명칭은 블랙리스트)을 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect

연산자: in-cidr-set | not-in-cidr-set

값: CIDR 목록

단위: 해당 사항 없음

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
}
```

연결 시도(aws:num-connection-attempts)

지정된 기간 동안 디바이스가 연결을 시도하는 횟수입니다.

각 디바이스가 연결을 시도한 최소 또는 최대 횟수를 지정하려면 이 지표를 사용합니다. 시도에 성공한 횟수와 실패한 횟수 모두 합산합니다.

호환 가능: Rules Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 연결 시도

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

```
{
  "name": "Connection Attempts",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
```

```
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **statisticalThreshold**를 사용하는 예제

```
{
  "name": "Connection Attempts",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect를 사용하는 예

```
{
  "name": "Connection attempts ML behavior",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": false
}
```

연결 해제(aws:num-disconnects)

지정된 기간 동안 디바이스가 AWS IoT에서 연결 해제되는 횟수입니다.

지정된 기간 동안 디바이스가 AWS IoT에서 연결 해제되는 최대 또는 최소 횟수를 지정하려면 이 지표를 사용합니다.

호환 가능: Rules Detect | ML Detect

연산자: less-than | less-than-equals | greater-than | greater-than-equals

값: 음수가 아닌 정수

단위: 연결 해제

기간: 음수가 아닌 정수 유효한 값은 300, 600, 900, 1800 또는 3600초입니다.

Example

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`**를 사용하는 예제

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
```

```
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect를 사용하는 예

```
{
  "name": "Disconnects ML behavior",
  "metric": "aws:num-disconnects",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

연결 해제 기간(aws:disconnect-duration)

디바이스가 AWS IoT에서 연결이 끊긴 상태로 유지되는 시간입니다.

이 지표를 사용하여 디바이스 AWS IoT에서 연결이 끊긴 상태로 유지되는 최대 기간을 지정합니다.

호환 가능: Rules Detect

연산자: less-than | less-than-equals

값: 음수가 아닌 정수(분)

Example

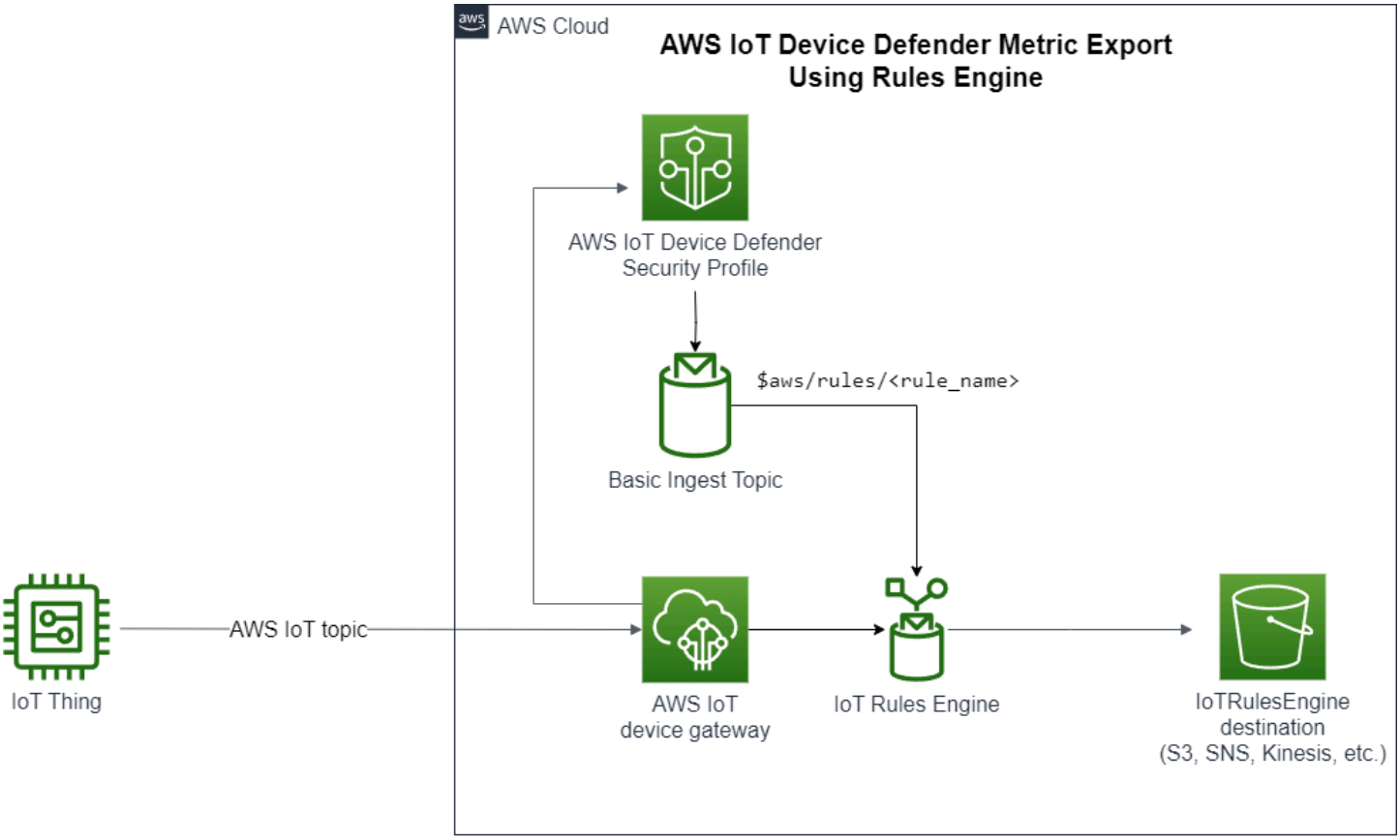
```
{
"name": "DisconnectDuration",
  "metric": "aws:disconnect-duration",
  "criteria": {
"comparisonOperator": "less-than-equals",
    "value": {
"count": 5
```

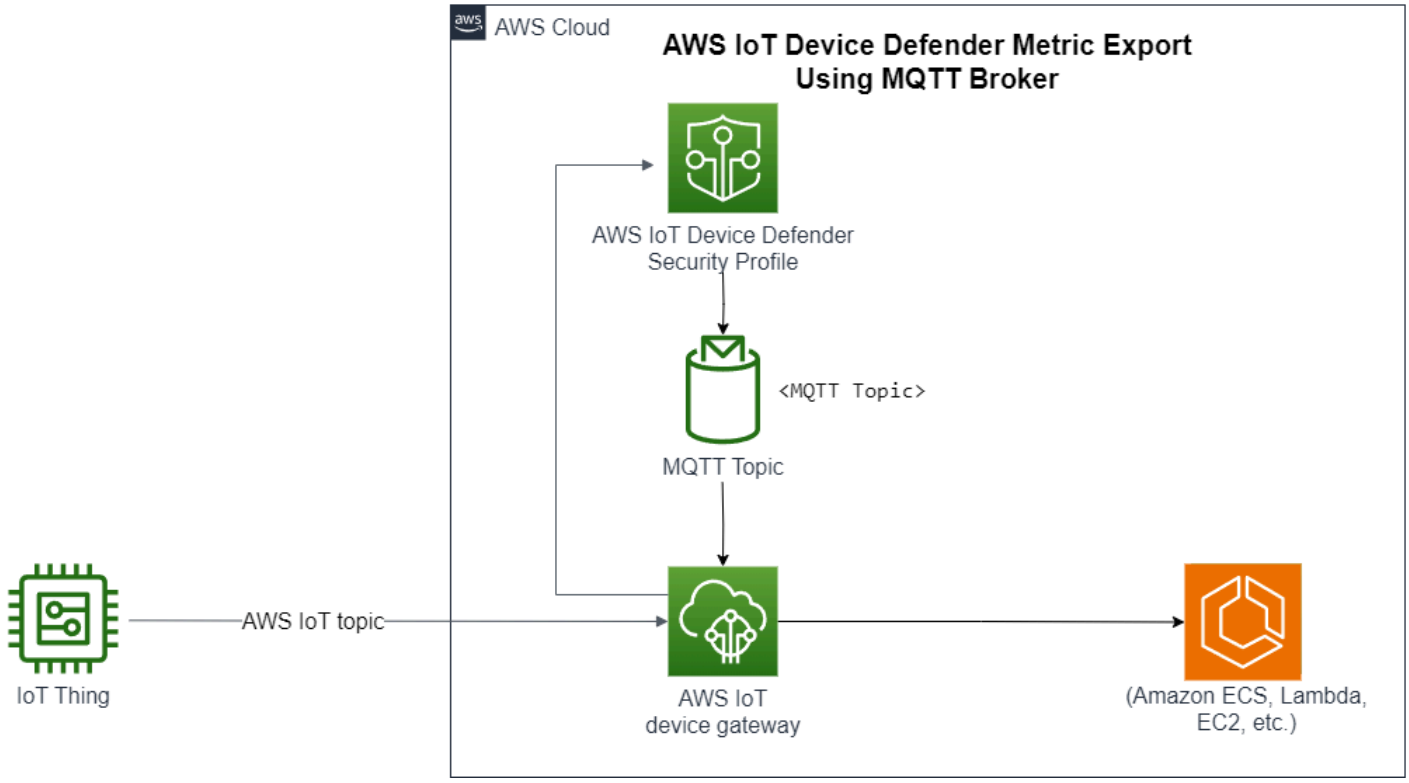
```
    }
  },
  "suppressAlerts": true
}
```

Detect 지표 내보내기

지표 내보내기를 사용하면 클라우드 측, 디바이스 측 또는 사용자 지정 지표를 AWS IoT Device Defender에서 내보내고 구성한 MQTT 주제에 게시할 수 있습니다. 이 기능은 Detect 지표의 대량 내보내기를 지원하므로 보다 효율적인 데이터 보고 및 분석뿐만 아니라 비용 관리에도 도움이 됩니다. MQTT 주제를 AWS IoT Rules 기본 수집 주제로 선택하거나 자체 MQTT 주제를 생성하여 구독할 수 있습니다. AWS IoT Device Defender 콘솔, API 또는 CLI를 사용하여 지표 내보내기를 구성할 수 있습니다. 이 기능은 AWS IoT Device Defender를 사용할 수 있는 모든 AWS 리전에서 사용할 수 있습니다.

다음 그림은 AWS IoT Device Defender가 지표를 내보내도록 구성하는 방법을 보여 줍니다. 첫 번째 다이어그램은 기본 수집 주제에 대한 지표 내보내기를 구성하는 방법을 보여 줍니다. 그런 다음 내보낸 지표를 AWS IoT Rules가 지원하는 다양한 대상으로 라우팅할 수 있습니다. 두 번째 다이어그램은 데이터를 MQTT 주제에 AWS IoT Device Defender 게시하도록 구성하는 방법을 보여 줍니다. 그러면 MQTT 클라이언트는 해당 주제를 구독합니다. Amazon Elastic 컨테이너 서비스, Lambda 또는 동일한 MQTT 주제를 구독하는 Amazon EC2 인스턴스에서 MQTT 클라이언트를 실행할 수 있습니다. AWS IoT Device Defender가 데이터를 게시할 때마다 MQTT 클라이언트는 데이터를 수신하여 처리합니다. 자세한 내용은 MQTT 주제를 참조하세요.


AWS Cloud
AWS IoT Device Defender Metric Export Using Rules Engine
AWS IoT Device Defender Security Profile
$aws/rules/<rule_name>
Basic Ingest Topic
IoT Thing
AWS IoT topic
AWS IoT device gateway
IoT Rules Engine
IoTRulesEngine destination (S3, SNS, Kinesis, etc.)


AWS Cloud
AWS IoT Device Defender Metric Export Using MQTT Broker
AWS IoT Device Defender Security Profile
<MQTT Topic>
MQTT Topic
IoT Thing
AWS IoT topic
AWS IoT device gateway
(Amazon ECS, Lambda, EC2, etc.)

지표 내보내기 탐지 작동 방식

보안 프로필을 설정할 때 내보낼 지표를 선택하고 MQTT 주제를 지정합니다. 또한 구성된 MQTT 주제에 메시지를 게시하는 데 필요한 권한을 AWS IoT Device Defender Detect에 부여하는 IAM 역할을 구성합니다. AWS IoT 규칙 기본 수집 MQTT 주제를 구성하고 내보낸 지표를 AWS IoT 규칙 지원 대상으로 보낼 수 있습니다. AWS IoT Rules 설정 및 구성에 대한 지침은 AWS IoT 개발자 안내서의 AWS IoT 규칙을 참조하세요.

AWS IoT Device Defender Detect는 구성된 각 지표에 대한 지표 값을 일괄 처리하여 구성된 MQTT 주제에 정기적으로 게시합니다. 메시지 바이트 크기 및 총 바이트 크기를 제외한 클라우드 측 지표는 일괄 처리 기간의 지표 값을 합산하여 집계됩니다. 사용자 지정 지표와 디바이스 측 지표는 집계되지 않습니다. 메시지 바이트 크기의 경우 내보내기 값은 일괄 처리 기간의 최소, 최대 및 총 바이트 크기입니다. 연결 해제 기간의 내보내기 값은 추적된 모든 디바이스의 연결 해제 시간(초) 입니다. 이는 1시간 간격으로 발생하며 연결 또는 연결 끊김 이벤트에서도 발생합니다. 연결된 디바이스 또는 연결 이벤트의 경우 값은 0이 됩니다. 클라우드 측 지표, 디바이스 측 지표 및 사용자 지정 지표에 대한 자세한 내용은 AWS IoT Device Defender 개발자 안내서의 다음 주제를 참조하세요.

- 사용자 지정 지표
- 클라우드 측 지표
- 디바이스 측 지표

AWS IoT Rules를 사용하여 일괄 처리된 지표를 서로 다른 대상으로 내보낼 수 있습니다. 지원되는 대상 목록은 AWS IoT 규칙 작업을 참조하세요. AWS IoT Rules 작업에 batchMode 옵션을 사용해 배치 처리된 내보내기 메시지 내의 개별 지표를 지원되는 대상으로 전송할 수 있습니다. 선호하는 AWS IoT Rules 대상이 `batchMode`를 지원하지 않는 경우 Lambda 또는 Kinesis Data Streams와 같은 중개 작업을 사용하여 배치 처리된 메시지 내에서 개별 지표를 전송할 수 있습니다.

지표 내보내기 스키마

일괄 지표 내보내기 데이터는 다음 스키마를 참조하세요.

```
{
 "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}"
 }
 }
 ]
}
```

지표 내보내기 탐지 요금

구성한 MQTT 주제에 클라우드 측, 디바이스 측 또는 사용자 지정 지표를 게시할 경우 내보내기 프로세스의 이 단계에 대한 요금이 발생하지 않습니다. 하지만 규칙 엔진이나 메시징을 사용하여 게시된 지표를 원하는 대상으로 전송하는 후속 단계에서 선택한 전송 방법에 따라 비용이 발생합니다. AWS IoT Device Defender는 일괄 처리된 지표를 여러 디바이스에 대한 지표 데이터가 포함된 단일 메시지

로 MQTT 주제에 게시하므로 비용을 관리하는 데 도움이 됩니다. 요금에 대한 자세한 내용은 AWS 요금 계산기를 참조하세요.

권한

이 섹션에는 AWS IoT Device Defender Detect 지표 내보내기를 관리하는 데 필요한 IAM 역할 및 정책을 설정하는 방법에 대한 정보가 있습니다. 자세한 내용은 IAM 사용 설명서를 참조하십시오.

AWS IoT Device Defender Detect에 MQTT 주제에 메시지를 게시할 권한을 부여합니다.

CreateSecurityProfile에서 지표 내보내기를 사용하려면 두 가지 정책, 즉 권한 정책과 신뢰 정책이 있는 IAM 역할을 지정해야 합니다. 권한 정책은 AWS IoT Device Defender가 지표가 포함된 메시지를 MQTT 주제에 게시할 권한을 부여합니다. 신뢰 정책은 AWS IoT Device Defender에 필요한 역할을 수임할 권한을 부여합니다.

권한 정책

```
{
  "Version":"2012-10-17",
  "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "iot:Publish"
        ],
        "Resource":[
          "arn:aws:iot:region:account-id:topic/your-topic-name"
        ]
    }
  ]
}
```

신뢰 정책

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

```
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

역할 정책 전달

사용자가 역할을 전달할 수 있도록 하는 IAM 권한 정책도 IAM 사용자에게 연결되어 있어야 합니다. AWS Service에 역할을 전달하기 위한 사용자 권한 부여를 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
          "iam:GetRole",
          "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/Role_To_Pass"
    }
  ]
}
```

AWS IoT 콘솔에서 지표 내보내기 탐지 설정

콘솔에서 지표 내보내기를 포함하는 새 보안 프로필을 생성 및 조회하고 편집합니다.

사전 조건

Detect 지표 내보내기를 설정하기 전에 다음과 같은 사전 조건을 충족해야 합니다.

- IAM 역할. IAM 역할 생성에 대한 자세한 내용은 IAM 사용 설명서의 IAM 역할 생성을 참조하세요.
- 올바른 권한을 가진 AWS Identity and Access Management(IAM) 사용자로 로그인할 수 있는 AWS 계정. AWS IoT Device Defender Detect 권한에 대한 자세한 내용은 AWS IoT Core개발자 안내서의 권한을 참조하세요.

지표 내보내기를 포함한 새 보안 프로필 생성(콘솔)

지표 동작 데이터를 내보내려면 먼저 지표 내보내기를 포함하도록 보안 프로필을 구성해야 합니다. 다음 절차는 Detect 지표 내보내기를 포함하는 규칙 기반 보안 프로필을 설정하는 방법을 자세히 설명합니다.

지표 내보내기가 포함된 새로운 보안 프로필 생성

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 감지, 보안 프로필을 확장합니다.
2. 보안 프로필 생성에서 규칙 기반 이상 탐지 프로필 생성을 선택합니다.
3. 보안 프로필 속성을 지정하려면 보안 프로필 이름을 입력하고 대상에서 이상의 대상으로 지정할 디바이스 그룹을 선택합니다. (선택 사항)AWS 리소스에 레이블을 지정하기 위한 설명과 태그를 포함하세요. [Next]를 선택합니다.
4. 지표에서는 디바이스 동작을 정의하는 지표를 선택합니다. 디바이스가 동작 기대치를 충족하지 못할 때 경고를 받을 동작 임계값을 정의할 수 있습니다.
5. 동작 이상에 대한 경고를 받으려면 알림 전송(지표 동작 정의)을 선택한 다음 동작 이름과 조건을 지정합니다. 지표를 경고 없이 보존하려면 알림 전송하지 않음(지표 유지)을 선택합니다. Next(다음)를 선택합니다.
6. 지표 내보내기를 구성하려면 지표 내보내기 활성화를 선택합니다.
7. 지표 데이터를 AWS IoT Core에 게시하는 데 사용할 MQTT 주제 이름을 입력합니다. AWS IoT에 “AWS IoT:Publish” 권한을 부여할 IAM 역할을 선택해 구성된 주제에 메시지를 게시합니다. 내보내려는 지표를 선택한 다음 다음을 선택합니다.

   > **Note**
   >
   > MQTT 주제 이름을 입력할 때는 슬래시를 사용하여 계층 정보를 나타냅니다. 예: `$AWS/rules/rule-name/`.

8. 디바이스가 설정된 동작을 위반할 때 AWS 콘솔로 경고를 보내도록 하려면 Amazon SNS 주제 및 IAM 역할을 선택하거나 생성합니다. [Next]를 선택합니다.
9. 구성을 검토한 후 다음을 선택합니다.

보안 프로필 세부 정보 보기 및 편집(콘솔)

보안 프로필 세부 정보를 보고 편집하려면

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 감지, 보안 프로필을 확장합니다.

2. 지표 내보내기를 포함하도록 생성한 보안 프로필을 선택한 다음 작업에서 편집을 선택합니다.
3. 대상에서 편집하려는 대상 디바이스 그룹을 선택한 후 다음을 선택합니다.
4. 지표 동작 구성을 편집하려면 알림(지표 동작 정의)를 선택한 다음 지표 동작이 충족될 때의 조건을 정의합니다. [Next]를 선택합니다.
5. 지표 내보내기 구성을 끄려면 지표 내보내기 비활성화를 선택합니다. [Next]를 선택합니다.
6. 디바이스가 설정된 동작을 위반할 때 AWS IoT 콘솔로 경고를 보내도록 Amazon SNS를 구성하려면 Amazon SNS 주제 및 IAM 역할을 선택하거나 생성합니다. [Next]를 선택합니다.
7. 구성을 검토한 후 다음을 선택합니다.

보안 프로필을 생성하여 지표 내보내기 활성화

`create-security-profile` 명령을 사용하여 보안 프로필을 생성하고 지표 내보내기를 활성화합니다.

지표 내보내기가 포함된 보안 프로필 생성

1. 지표 내보내기를 활성화하고 Detect에서 해당 지표를 내보내야 하는지 여부를 표시하려면 `Behavior` 및 `AdditionalMetricsToRetainV2` 모두에서 `exportMetric` 값을 True로 설정합니다.
2. `MetricsExportConfig`의 값을 포함하세요. 이렇게 하면 지표 내보내기에 필요한 MQTT 주제 및 Amazon 리소스 이름(ARN)이 지정됩니다.

> **Note**
>
> AWS IoT Device Defender Detect가 메시지를 게시할 수 있도록 `mqttTopic`을 포함합니다. 역할 ARN에는 MQTT 메시지를 게시할 권한이 있으며, 이후에는 AWS IoT Device Defender Detect가 역할을 맡아 사용자를 대신하여 메시지를 게시할 수 있습니다.

```
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
```

출력:

```
{
    "securityProfileName": "CreateSecurityProfileWithMetricsExport",
    "securityProfileArn": "arn:aws:iot:us-east-1:123456789012:securityprofile/
CreateSecurityProfileWithMetricsExport"
}
```

보안 프로필을 업데이트하여 지표 내보내기 활성화(CLI)

`update-security-profile` 명령을 사용하여 기존 보안 프로필을 업데이트하고 지표 내보내기를 활성화합니다.

보안 프로필을 업데이트하여 지표 내보내기를 활성화하려면

1. 지표 내보내기를 활성화하고 Detect에서 해당 지표를 내보내야 하는지 여부를 표시하려면 `Behavior` 및 `AdditionalMetricsToRetainV2` 모두에서 `exportMetric` 값을 True로 설정합니다.
2. `MetricsExportConfig`의 값을 포함하세요. 이렇게 하면 지표 내보내기에 필요한 MQTT 주제 및 Amazon 리소스 이름(ARN)이 지정됩니다.

> **Note**
>
> AWS IoT Device Defender Detect가 메시지를 게시할 수 있도록 `mqttTopic`을 포함합니다. 역할 ARN에는 MQTT 메시지를 게시할 권한이 있으며, 이후에는 AWS IoT Device Defender Detect가 역할을 맡아 사용자를 대신하여 메시지를 게시할 수 있습니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

보안 프로필을 업데이트하여 지표 내보내기 비활성화(CLI)

`update-security-profile` 명령을 사용하여 기존 보안 프로필을 업데이트하고 지표 내보내기를 비활성화합니다.

보안 프로필을 업데이트하여 지표 내보내기를 비활성화하려면.

- 보안 프로필을 업데이트하고 지표 내보내기 구성을 제거하려면 `--delete-metrics-export-config` 명령을 사용합니다.

```
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
```

출력:

```
{
    "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"
```

```
}
```

자세한 정보는 AWS IoT 개발자 안내서의 명령 탐지를 참조하세요.

지표 내보내기 CLI 명령

다음 CLI 명령을 사용하여 탐지 지표 내보내기를 만들고 관리할 수 있습니다.

- CreateSecurityProfile
- UpdateSecurityProfile
- DescribeSecurityProfile

지표 내보내기 API 작업

다음 API 명령을 사용하여 Detect 지표 내보내기를 만들고 관리할 수 있습니다.

- CreateSecurityProfile
- UpdateSecurityProfile
- DescribeSecurityProfile

차원을 사용하여 보안 프로파일의 지표 범위 지정

차원은 보안 프로파일의 지표 및 동작에 대한 보다 정확한 데이터를 얻기 위해 정의할 수 있는 속성입니다. 필터로 사용되는 값이나 패턴을 제공하여 범위를 정의합니다. 예를 들어 “data/bulb/+/activity”와 같이 특정 값과 일치하는 MQTT 주제에만 지표를 적용하는 주제 필터 차원을 정의할 수 있습니다. 보안 프로파일에서 사용할 수 있는 차원 정의에 대한 자세한 내용은 CreateDimension을 참조하세요.

차원 값은 MQTT 와일드카드를 지원합니다. MQTT 와일드카드를 사용하면 여러 주제를 동시에 구독할 수 있습니다. 와일드카드에는 단일 레벨(+) 및 다중 레벨((#)의 두 가지 종류가 있습니다. 예를 들어 차원 값 `Data/bulb/+/activity`는 +와 동일한 레벨에 있는 모든 주제와 일치하는 구독을 생성합니다. 차원 값은 MQTT 클라이언트 ID 대체 변수 ${iot:ClientId}도 지원합니다.

TOPIC_FILTER 유형의 차원은 다음과 같은 클라우드 측 지표 집합과 호환됩니다.

- 권한 부여 실패 횟수
- 메시지 바이트 크기
- 수신된 메시지 수

- 전송된 메시지 수
- 소스 IP 주소(규칙 감지에만 사용 가능)

콘솔에서 차원을 사용하는 방법

차원을 생성하여 보안 프로파일 동작에 적용하려면

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 탐지를 확장한 다음 보안 프로필을 선택합니다.
2. 보안 프로필 페이지에서 보안 프로필 생성을 선택하고 규칙 기반 이상 탐지 프로필 생성을 선택합니다. 또는 기존 규칙 기반 보안 프로필에 차원을 적용하려면 보안 프로필을 선택하고 편집을 선택합니다.
3. 보안 프로필 속성 지정 페이지에서 보안 프로필의 이름을 입력합니다.
4. 이상 현상의 대상으로 삼을 디바이스 그룹을 선택합니다.
5. 다음을 선택합니다.
6. 지표 동작 구성 페이지의 지표 유형에서 클라우드 측 지표 차원 중 하나를 선택합니다.
7. 지표 동작의 경우 알림 전송(지표 동작 정의)을 선택하여 예상 지표 동작을 정의합니다.
8. 비정상적인 디바이스 동작에 대해 알림을 받고자 하는 시기를 선택합니다.
9. 다음을 선택합니다.
10. 보안 프로필 구성을 살펴본 후 생성을 선택합니다.

경보를 보는 방법

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 탐지를 확장한 다음 경보를 선택합니다.
2. 사물 이름 열에서 경보를 유발한 원인에 대한 정보를 확인할 사물을 선택합니다.

차원을 보고 업데이트하려면

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 탐지를 확장한 다음 차원을 선택합니다.
2. 차원을 선택하고 편집을 선택합니다.
3. 차원을 편집하고 업데이트를 선택합니다.

차원을 삭제하려면

1. AWS IoT 콘솔을 엽니다. 탐색 창에서 보안, 탐지를 확장한 다음 차원을 선택합니다.

2. 차원을 삭제하기 전에 해당 차원을 참조하는 지표 동작을 삭제해야 합니다. 보안 프로필 열을 선택하여 차원이 보안 프로필에 연결되지 않았는지 확인합니다. 차원이 보안 프로필에 연결된 경우 왼쪽의 보안 프로필 페이지를 열고 차원이 연결된 보안 프로필을 편집합니다. 그런 다음 동작을 삭제할 수 있습니다. 다른 차원을 삭제하려면 이 섹션의 단계를 수행합니다.
3. 차원을 선택하고 삭제를 선택합니다.
4. 차원 이름을 입력하여 확인하고 삭제를 선택합니다.

AWS CLI에서 차원을 사용하는 방법

차원을 생성하여 보안 프로파일 동작에 적용하려면

1. 차원을 생성한 다음 보안 프로파일에 연결합니다. CreateDimension 명령을 사용하여 차원을 생성합니다.

```
aws iot create-dimension \
  --name TopicFilterForAuthMessages \
  --type TOPIC_FILTER \
  --string-values device/+/auth
```

이 명령의 출력은 다음과 같습니다.

```
{
    "arn": "arn:aws:iot:us-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
    "name": "TopicFilterForAuthMessages"
}
```

2. UpdateSecurityProfile을 사용하여 기존 보안 프로파일에 차원을 추가하거나 CreateSecurityProfile을 사용하여 새 보안 프로파일에 차원을 추가합니다. 다음 예제에서는 `TopicFilterForAuthMessages`에 대한 메시지가 128바이트 미만인지 확인하고 미인증(non-auth) 주제에 전송된 메시지 수를 유지하는 새 보안 프로파일을 생성합니다.

```
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\"}}]"
```

이 명령의 출력은 다음과 같습니다.

```
{
    "securityProfileArn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice",
    "securityProfileName": "ProfileForConnectedDevice"
}
```

또한 파라미터를 명령줄 파라미터 값으로 입력하는 대신 파일에서 로드하여 시간을 절약할 수 있습니다. 자세한 내용은 파일에서 AWS CLI 파라미터 로드를 참조하세요. 다음은 확장된 JSON 형식의 `behavior` 파라미터를 보여줍니다.

```
[
  {
    "criteria": {
      "comparisonOperator": "less-than",
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "value": {
        "count": 128
      }
    },
    "metric": "aws:message-byte-size",
    "metricDimension": {
      "dimensionName:": "TopicFilterForAuthMessages"
    },
    "name": "CellularBandwidth"
  }
]
```

또는 다음 예와 같이 ML과 함께 차원을 사용하여 CreateSecurityProfile을 사용합니다.

```
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
```

차원이 있는 보안 프로파일을 보려면

- ListSecurityProfiles 명령을 사용하여 특정 차원이 있는 보안 프로파일을 봅니다.

```
aws iot list-security-profiles \
  --dimension-name TopicFilterForAuthMessages
```

이 명령의 출력은 다음과 같습니다.

```
{
    "securityProfileIdentifiers": [
        {
            "name": "ProfileForConnectedDevice",
            "arn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice"
        }
    ]
}
```

차원을 업데이트하려면

- UpdateDimension 명령을 사용하여 차원을 업데이트합니다.

```
aws iot update-dimension \
  --name TopicFilterForAuthMessages \
  --string-values device/${iot:ClientId}/auth
```

이 명령의 출력은 다음과 같습니다.

```
{
    "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"
}
```

차원을 삭제하려면

1. 차원을 삭제하려면 먼저 연결되어 있는 보안 프로파일에서 차원을 분리합니다. ListSecurityProfiles 명령을 사용하여 특정 차원이 있는 보안 프로파일을 봅니다.
2. 보안 프로파일에서 차원을 제거하려면 UpdateSecurityProfile 명령을 사용합니다. 차원을 제외한 유지하려는 모든 정보를 입력합니다.

```
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}}]"
```

이 명령의 출력은 다음과 같습니다.

```
{
  "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. 차원을 분리한 후 DeleteDimension 명령을 사용하여 차원을 삭제합니다.

```
aws iot delete-dimension \
  --name TopicFilterForAuthMessages
```

권한

이 섹션에는 AWS IoT Device Defender Detect를 관리하는 데 필요한 IAM 역할 및 정책을 설정하는 방법에 대한 정보가 있습니다. 자세한 내용은 IAM 사용 설명서를 참조하십시오.

AWS IoT Device Defender Detect에 SNS 주제에 경보를 게시할 권한을 부여합니다.

CreateSecurityProfile에서 `alertTargets` 파라미터를 사용하려면 두 가지 정책, 즉 권한 정책과 신뢰 정책이 있는 IAM 역할을 지정해야 합니다. 권한 정책은 SNS 주제에 알림을 게시할 권한을 AWS IoT Device Defender에 부여합니다. 신뢰 정책은 AWS IoT Device Defender에 필요한 역할을 수임할 권한을 부여합니다.

권한 정책

```
{
  "Version":"2012-10-17",
  "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "sns:Publish"
        ],
        "Resource":[
          "arn:aws:sns:region:account-id:your-topic-name"
        ]
    }
  ]
}
```

신뢰 정책

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

역할 정책 전달

사용자가 역할을 전달할 수 있도록 하는 IAM 권한 정책도 IAM 사용자에게 연결되어 있어야 합니다. AWS Service에 역할을 전달하기 위한 사용자 권한 부여를 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
          "iam:GetRole",
          "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/Role_To_Pass"
    }
  ]
}
```

Detect 명령

이 섹션의 감지(Detect) 명령을 사용하여 ML Detect 또는 Rules Detect 보안 프로파일을 구성하고, 손상된 디바이스를 나타낼 수 있는 비정상적인 동작을 식별하고 모니터링할 수 있습니다.

DetectMitigation 작업 명령

감지(Detect) 실행 시작 및 관리
CancelDetectMitigationActionsTask
DescribeDetectMitigationActionsTask
ListDetectMitigationActionsTasks
StartDetectMitigationActionsTask
ListDetectMitigationActionsExecutions

차원(Dimension) 작업 명령

차원(Dimension) 실행 시작 및 관리
CreateDimension
DescribeDimension
ListDimensions
DeleteDimension
UpdateDimension

CustomMetric 작업 명령

CustomMetric 실행 시작 및 관리
CreateCustomMetric
UpdateCustomMetric
DescribeCustomMetric
ListCustomMetrics
DeleteCustomMetric

보안 프로파일 작업 명령

보안 프로파일 실행 시작 및 관리
CreateSecurityProfile
AttachSecurityProfile
DetachSecurityProfile
DeleteSecurityProfile
DescribeSecurityProfile

보안 프로파일 실행 시작 및 관리
ListTargetsForSecurityProfile
UpdateSecurityProfile
ValidateSecurityProfileBehaviors
ListSecurityProfilesForTarget

경보(Alarm) 작업 명령

경보 및 대상 관리
ListActiveViolations
ListViolationEvents
PutVerificationStateOnViolation

ML Detect 작업 명령

ML 모델 훈련 데이터 열거
GetBehaviorModelTrainingSummaries

AWS IoT Device Defender detect 사용 방법

1. 클라우드 측 지표를 통해서만 AWS IoT Device Defender Detect를 사용할 수 있지만, 디바이스에서 보고된 지표를 사용할 계획인 경우 먼저 AWS IoT가 연결된 디바이스 또는 디바이스 게이트웨이에 AWS IoT SDK를 배포해야 합니다. 자세한 내용은 디바이스에서 지표 전송 단원을 참조하세요.
2. 동작을 정의하고 경보를 생성하기 전에 디바이스에서 생성되는 지표 확인을 고려합니다. AWS IoT는 디바이스에서 지표를 수집할 수 있으므로 먼저 디바이스 그룹 또는 계정의 모든 디바이스에 대해 정상적이거나 비정상적인 동작을 식별할 수 있습니다. CreateSecurityProfile를 사용하되 관심 있는 `additionalMetricsToRetain`만 지정합니다. 이때 `behaviors`을(를) 지정하지 마세요.

AWS IoT 콘솔을 사용하여 디바이스 지표를 살펴보고 디바이스의 일반적인 동작을 구성하는 항목을 확인합니다.

3. 보안 프로파일을 위한 동작 집합을 생성합니다. 동작에는 디바이스 그룹 또는 계정의 모든 디바이스에 대한 정상 동작을 지정하는 지표가 포함됩니다. 자세한 정보와 예제를 보려면 클라우드 측 지표 단원과 Device-side metrics 단원을 참조하세요. 동작 집합을 생성했으면 ValidateSecurityProfileBehaviors에 따라 검증할 수 있습니다.
4. CreateSecurityProfile 작업을 사용하여 동작이 포함된 보안 프로파일을 생성합니다. 디바이스에서 동작을 위반한 경우 `alertTargets` 파라미터를 사용하여 대상(SNS 주제)으로 경보를 전송할 수 있습니다. (SNS를 통해 경보를 전송할 경우 이러한 알림 전송이 AWS 계정의 SNS 주제 할당량에 포함됨을 유의하세요.) 대규모 위반이 SNS 주제 할당량을 초과할 수 있습니다. CloudWatch 지표를 사용하여 위반 여부를 검사할 수도 있습니다. 자세한 내용은 AWS IoT Core 개발자 설명서의 Amazon CloudWatch를 사용하여 AWS IoT 경보 및 지표 모니터링을 참조하세요.
5. AttachSecurityProfile 작업을 사용하여 디바이스 그룹(사물 그룹), 계정에 등록된 모든 사물, 등록되지 않은 모든 사물 또는 모든 디바이스에 보안 프로파일을 연결합니다. AWS IoT Device Defender Detect가 비정상적인 동작을 검사하기 시작하고, 동작 위반이 감지된 경우 경보를 전송합니다. 예를 들어 계정의 사물 레지스트리에 없는 모바일 디바이스와 상호 작용할 것으로 예상되는 경우 등록되지 않은 모든 사물에 보안 프로파일을 연결할 수 있습니다. 요건에 맞게 서로 다른 디바이스 그룹에 대해 서로 다른 동작 집합을 정의할 수 있습니다.

   디바이스 그룹에 보안 프로파일을 연결하려면 해당 그룹이 포함된 사물 그룹의 ARN을 지정해야 합니다. 사물 그룹 ARN의 형식은 다음과 같습니다.

   ```
   arn:aws:iot:region:account-id:thinggroup/thing-group-name
   ```

   AWS 계정에 등록된 모든 사물에 보안 프로파일을 연결하려면(등록되지 않은 사물 무시) 다음 형식으로 ARN을 지정해야 합니다.

   ```
   arn:aws:iot:region:account-id:all/registered-things
   ```

   등록되지 않은 모든 사물에 보안 프로파일을 연결하려면 다음 형식으로 ARN을 지정해야 합니다.

   ```
   arn:aws:iot:region:account-id:all/unregistered-things
   ```

   모든 디바이스에 보안 프로파일을 연결하려면 다음 형식으로 ARN을 지정해야 합니다.

```
arn:aws:iot:region:account-id:all/things
```

6. ListActiveViolations 작업을 통해 계속 위반을 추적하여 지정된 보안 프로파일 또는 대상 디바이스에 대해 감지된 위반을 확인할 수도 있습니다.

   ListViolationEvents 작업을 사용하여 지정된 기간 동안 감지된 위반을 확인합니다. 보안 프로파일, 디바이스 또는 경보 확인 상태를 기준으로 이러한 결과를 필터링할 수 있습니다.
7. PutVerificationStateOnViolation 작업을 사용하여 확인 상태를 표시하고 해당 확인 상태에 대한 설명을 제공하여 경보를 확인, 구성 및 관리할 수 있습니다.
8. 디바이스가 정의된 동작을 너무 자주 또는 가끔 위반하는 경우, 동작 정의를 세부 조정해야 합니다.
9. 설정한 보안 프로파일과 모니터링 중인 디바이스를 검토하려면 ListSecurityProfiles, ListSecurityProfilesForTarget 및 ListTargetsForSecurityProfile 작업을 수행합니다.

   DescribeSecurityProfile 작업을 사용하여 보안 프로파일에 대한 추가 세부 정보를 가져옵니다.
10. 보안 프로파일을 업데이트하려면 UpdateSecurityProfile 작업을 사용합니다. DetachSecurityProfile 작업을 사용하여 계정 또는 대상 사물 그룹에서 보안 프로파일을 분리합니다. DeleteSecurityProfile 작업을 사용하여 보안 프로파일을 완전히 삭제합니다.

완화 작업

AWS IoT Device Defender을(를) 사용하여 감사 결과 또는 감지 경보에서 발견한 문제를 완화하는 작업을 수행합니다.

> **Note**
> 금지된 감사 결과에 대해서는 완화 작업이 수행되지 않습니다. 감사 결과 금지에 대한 자세한 내용은 감사 결과 금지 단원을 참조하세요.

감사 완화 작업

AWS IoT Device Defender에서는 여러 감사 검사에 대해 미리 정의된 작업을 제공합니다. AWS 계정에 대해 이러한 작업을 구성한 다음 일련의 결과에 적용합니다. 결과는 다음 중 하나일 수 있습니다.

- 모든 감사 결과. 이 옵션은 AWS IoT 콘솔에서 사용할 수 있으며 AWS CLI로도 사용할 수 있습니다.
- 개별 결과의 목록. 이 옵션은 AWS CLI로만 사용할 수 있습니다.
- 감사 결과 중 필터링된 집합.

다음 표에는 감사 검사의 유형과 각 유형에 지원되는 완화 작업이 나열되어 있습니다.

감사 검사 - 완화 작업 매핑

감사 검사	지원되는 완화 작업
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
DEVICE_CERTIFICATE_SHARED_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP

감사 검사	지원되는 완화 작업
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
CA_CERTIFICATE_KEY_QUALITY_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS

감사 검사	지원되는 완화 작업
IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK	PUBLISH_FINDING_TO_SNS

모든 감사 검사는 Amazon SNS에 대한 감사 결과 게시를 지원하므로 알림에 따라 사용자 지정 작업을 수행할 수 있습니다. 각 감사 검사 유형마다 추가적인 완화 작업을 지원할 수 있습니다.

REVOKED_CA_CERT_CHECK

- 인증서 상태를 변경하여 AWS IoT에서 비활성으로 표시합니다.

DEVICE_CERTIFICATE_SHARED_CHECK

- 디바이스 인증서 상태를 변경하여 AWS IoT에서 비활성으로 표시합니다.
- 해당 인증서를 사용하는 디바이스를 사물 그룹에 추가합니다.

UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- 추가로 지원되는 작업이 없습니다.

AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- 추가로 지원되는 작업이 없습니다.

IOT_POLICY_OVERLY_PERMISSIVE_CHECK

- 빈 AWS IoT 정책 버전을 추가하여 권한을 제한합니다.

IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK

- AWS IoT 정책의 잠재적 구성 오류를 식별합니다.

CA_CERT_APPROACHING_EXPIRATION_CHECK

- 인증서 상태를 변경하여 AWS IoT에서 비활성으로 표시합니다.

CONFLICTING_CLIENT_IDS_CHECK

- 추가로 지원되는 작업이 없습니다.

DEVICE_CERT_APPROACHING_EXPIRATION_CHECK

- 디바이스 인증서 상태를 변경하여 AWS IoT에서 비활성으로 표시합니다.
- 해당 인증서를 사용하는 디바이스를 사물 그룹에 추가합니다.

DEVICE_CERTIFICATE_KEY_QUALITY_CHECK

- 디바이스 인증서 상태를 변경하여 AWS IoT에서 비활성으로 표시합니다.
- 해당 인증서를 사용하는 디바이스를 사물 그룹에 추가합니다.

CA_CERTIFICATE_KEY_QUALITY_CHECK

- 인증서 상태를 변경하여 AWS IoT에서 비활성으로 표시합니다.

REVOKED_DEVICE_CERT_CHECK

- 디바이스 인증서 상태를 변경하여 AWS IoT에서 비활성으로 표시합니다.
- 해당 인증서를 사용하는 디바이스를 사물 그룹에 추가합니다.

LOGGING_DISABLED_CHECK

- 로깅을 활성화합니다.

AWS IoT Device Defender은(는) 감사 결과에 대해 다음과 같은 유형의 완화 작업을 지원합니다.

작업 유형	참고
ADD_THINGS_TO_THING_GROUP	디바이스를 추가할 그룹을 지정합니다. 사물이 속할 수 있는 최대 그룹 수를 초과한 경우 동적 그룹 하나 이상의 회원 자격을 재정의해야 하는지 여부도 지정합니다.
ENABLE_IOT_LOGGING	로깅 수준과 로깅 권한을 가진 역할을 지정합니다. `DISABLED`의 로깅 수준은 지정할 수 없습니다.
PUBLISH_FINDING_TO_SNS	결과를 게시해야 하는 주제를 지정합니다.
REPLACE_DEFAULT_POLICY_VERSION	템플릿 이름을 지정합니다. 정책 버전을 기본 또는 빈 정책으로 바꿉니다. 현재 `BLANK_POLICY` 값만 지원됩니다.
UPDATE_CA_CERTIFICATE	CA 인증서의 새로운 상태를 지정합니다. 현재 `DEACTIVATE` 값만 지원됩니다.
UPDATE_DEVICE_CERTIFICATE	디바이스 인증서의 새로운 상태를 지정합니다. 현재 `DEACTIVATE` 값만 지원됩니다.

감사 중 문제가 발견되는 경우에 표준 작업을 구성해 두면 이러한 문제에 일관되게 대응할 수 있습니다. 이렇게 정의된 완화 작업을 사용하면 문제를 더욱 빠르게 처리하고 사람에 의해 발생하는 오류 가능성을 줄일 수 있습니다.

> **Important**
>
> 인증서를 변경하거나 새로운 사물 그룹에 사물을 추가하거나 정책을 바꾸는 완화 작업을 수행하면 디바이스와 애플리케이션에 영향이 있을 수 있습니다. 예를 들어 디바이스를 연결할 수 없는 경우가 있을 수 있습니다. 완화 작업을 수행하기 전에 이 작업이 미칠 영향을 고려하세요. 디바이스와 애플리케이션이 정상적으로 작동하려면 문제를 바로잡기 위한 다른 작업을 수행해야 할 수 있습니다. 예를 들어 업데이트된 디바이스 인증서를 제공해야 할 수 있습니다. 완화 작업은 위험을 신속하게 제한하는 데 도움이 될 수 있으나 근본 문제를 해결하기 위해서는 여전히 수정 작업을 수행해야 합니다.

디바이스 인증서의 재활성화 같은 일부 작업은 수동으로만 수행할 수 있습니다. AWS IoT Device Defender는 이미 수행된 완화 작업을 자동으로 롤백하는 메커니즘을 제공하지 않습니다.

완화 작업 감지

AWS IoT Device Defender은(는) 감지 경보에 대해 다음과 같은 유형의 완화 작업을 지원합니다.

작업 유형	참고
ADD_THINGS_TO_THING_GROUP	디바이스를 추가할 그룹을 지정합니다. 사물이 속할 수 있는 최대 그룹 수를 초과한 경우 동적 그룹 하나 이상의 회원 자격을 재정의해야 하는지 여부도 지정합니다.

완화 작업을 정의하고 관리하는 방법

AWS IoT 콘솔이나 AWS CLI을(를) 사용하여 AWS 계정의 완화 작업을 정의하고 관리할 수 있습니다.

완화 작업 생성

정의한 각 완화 작업은 사전 정의된 작업 유형과 계정 전용 파라미터의 조합으로 이루어져 있습니다.

AWS IoT 콘솔을 사용하여 완화 작업을 생성하려면

1. AWS IoT 콘솔의 Mitigation actions(완화 작업) 페이지를 엽니다.

2. Mitigation actions(완화 작업) 페이지에서 Create(생성)를 선택합니다.
3. Create a new mitigation action(새 완화 작업 생성) 페이지의 Action name(작업 이름)에서 완화 작업에 고유한 이름을 입력합니다.
4. 작업 유형에서 정의할 작업의 유형을 지정합니다.
5. Permissions(권한)에서 작업이 적용되는 권한의 IAM 역할을 선택합니다.
6. 작업 유형마다 다른 파라미터 집합을 요청합니다. 작업의 파라미터를 입력합니다. 예를 들어, 사물 그룹에 사물 추가 작업 유형을 선택한 경우 대상 그룹을 선택한 다음 Override dynamic groups(동적 그룹 재정의)를 선택하거나 지웁니다.
7. Create(생성)를 선택하여 완화 작업을 AWS 계정에 저장합니다.

AWS CLI를 사용하여 완화 작업을 생성하려면

- CreateMitigationAction 명령을 사용하여 완화 작업을 생성합니다. 작업을 감사 결과에 적용할 때는 작업에 붙인 고유한 이름이 사용됩니다. 의미 있는 이름을 선택하세요.

AWS IoT 콘솔을 사용하여 완화 작업을 확인하고 수정하려면

1. AWS IoT 콘솔의 Mitigation actions(완화 작업) 페이지를 엽니다.

   Mitigation Actions(완화 작업) 페이지에 AWS 계정에 대해 정의된 모든 완화 작업의 목록이 표시됩니다.
2. 변경하려는 완화 작업의 작업 이름 링크를 선택합니다.
3. Edit(편집)을 선택하여 완화 작업을 변경합니다. 완화 작업의 이름은 작업을 식별하는 데 사용되므로 이름을 변경할 수 없습니다.
4. Update(업데이트)를 선택하여 완화 작업에 대한 변경 사항을 AWS 계정에 저장합니다.

AWS CLI를 사용하여 완화 작업을 나열하려면

- ListMitigationAction 명령을 사용하여 완화 작업을 나열합니다. 완화 작업을 변경하거나 삭제하려면 이름을 기록해 둡니다.

AWS CLI를 사용하여 완화 작업을 업데이트하려면

- UpdateMitigationAction 명령을 사용하여 완화 작업을 변경합니다.

AWS IoT 콘솔을 사용하여 완화 작업을 삭제하려면

1. AWS IoT 콘솔의 Mitigation actions(완화 작업) 페이지를 엽니다.

   완화 작업(Mitigation actions) 페이지에 AWS 계정에 대해 정의된 모든 완화 작업이 표시됩니다.
2. 삭제하려는 완화 작업을 선택한 다음 Delete(삭제)를 선택합니다.
3. Are you sure you want to delete(삭제하시겠습니까?) 창에서 Delete(삭제)를 선택합니다.

AWS CLI 콘솔을 사용하여 완화 작업을 삭제하려면

- UpdateMitigationAction 명령을 사용하여 완화 작업을 변경합니다.

AWS IoT 콘솔을 사용하여 완화 작업의 세부 정보를 확인하려면

1. AWS IoT 콘솔의 Mitigation actions(완화 작업) 페이지를 엽니다.

   완화 작업(Mitigation actions) 페이지에 AWS 계정에 대해 정의된 모든 완화 작업이 표시됩니다.
2. 확인하려는 완화 작업의 작업 이름 링크를 선택합니다.

AWS CLI를 사용하여 완화 작업의 세부 정보를 확인하려면

- DescribeMitigationAction 명령을 사용하여 완화 작업의 세부 정보를 봅니다.

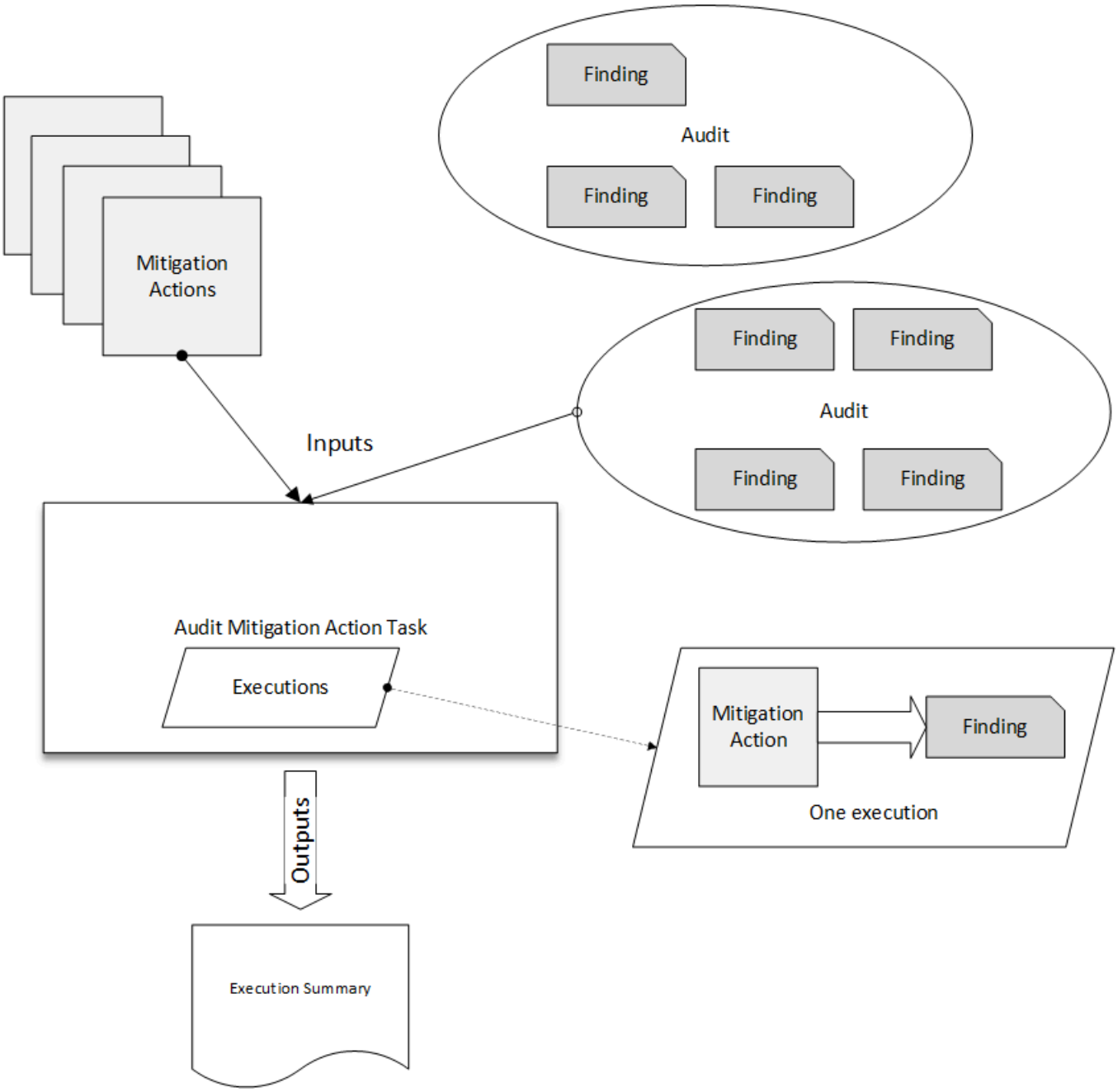
완화 작업 적용

완화 작업 집합을 정의한 후 이러한 작업을 감사 결과에 적용합니다. 작업을 적용하면 감사 완화 작업을 시작합니다. 결과 집합과 적용하는 작업에 따라 이 작업을 완료하는 데 다소 시간이 소요될 수 있습니다. 예를 들어 인증서가 만료된 인증서의 대규모 풀이 있는 경우 모든 인증서를 비활성화하거나 디바이스를 격리 그룹으로 옮기는 데 시간이 소요될 수 있습니다. 로깅 활성화와 같은 다른 작업은 빠르게 완료됩니다.

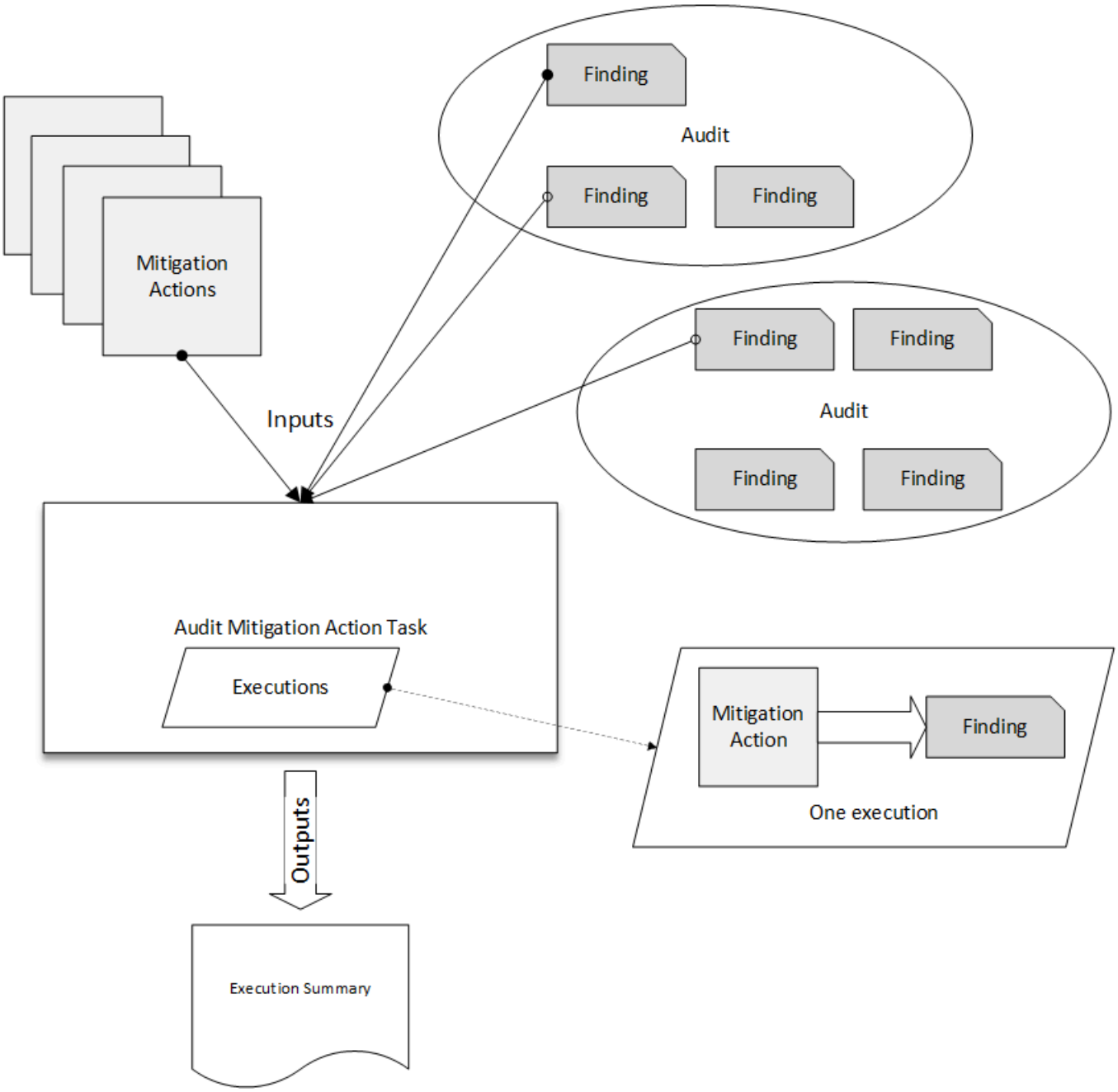
작업 실행 목록을 확인하고 아직 완료되지 않은 실행을 취소할 수 있습니다. 취소된 작업 실행 중 이미 수행한 작업은 롤백되지 않습니다. 여러 작업을 결과 집합에 적용하던 중 작업 중 하나가 실패한 경우 해당 결과는 후속 작업을 건너 뜁니다(다른 결과에는 적용됨) 결과의 작업 상태는 FAILED입니다. 하나 이상의 작업이 결과에 적용될 때 실패한 경우 `taskStatus`는 FAILED로 설정됩니다. 작업은 지정된 순서대로 적용됩니다.

각 작업 실행 시 작업 집합이 대상에 적용됩니다. 대상은 결과 목록이거나 모든 감사 결과일 수 있습니다.

다음 다이어그램은 하나의 감사에서 모든 결과를 가져와서 이 결과에 작업 집합을 적용하는 감사 완화 작업을 어떻게 정의하는지 보여줍니다. 한 번 실행 시 하나의 결과에 하나의 결과가 적용됩니다. 감사 완화 작업에서 실행 요약이 출력됩니다.



다음 다이어그램은 하나 이상의 감사에서 개별 결과 목록을 가져와서 이 결과에 작업 집합을 적용하는 감사 완화 작업을 어떻게 정의하는지 보여줍니다. 한 번 실행 시 하나의 결과에 하나의 결과가 적용됩니다. 감사 완화 작업에서 실행 요약이 출력됩니다.



AWS IoT 콘솔 또는 AWS CLI를 사용하여 완화 정책을 적용할 수 있습니다.

AWS IoT 콘솔에서 작업 실행으로 완화 정책을 적용하려면

1. AWS IoT 콘솔의 Audit results(감사 결과) 페이지를 엽니다.

2. 작업을 적용할 감사의 이름을 선택합니다.
3. Start mitigation actions(완화 작업 시작)를 선택합니다. 모든 검사가 규정을 준수하는 경우 이 버튼을 사용할 수 없습니다.
4. Start a new mitigation action(새 완화 작업 시작)에서 작업 이름은 감사 ID로 기본 설정되지만 더 의미 있는 이름으로 변경할 수 있습니다.
5. 감사에 규정을 준수하지 않는 결과가 하나 이상 있는 각 검사 유형의 경우 적용할 작업을 하나 이상 선택할 수 있습니다. 검사 유형에 대해 유효한 작업만 표시됩니다.

> **Note**
>
> AWS 계정에 대한 작업을 구성하지 않은 경우 해당 작업 목록은 비어 있습니다. Create mitigation action(완화 작업 생성) 링크를 선택하여 하나 이상의 완화 작업을 생성할 수 있습니다.

6. 적용하려는 모든 작업을 지정했으면 Start task(작업 시작)를 선택합니다.

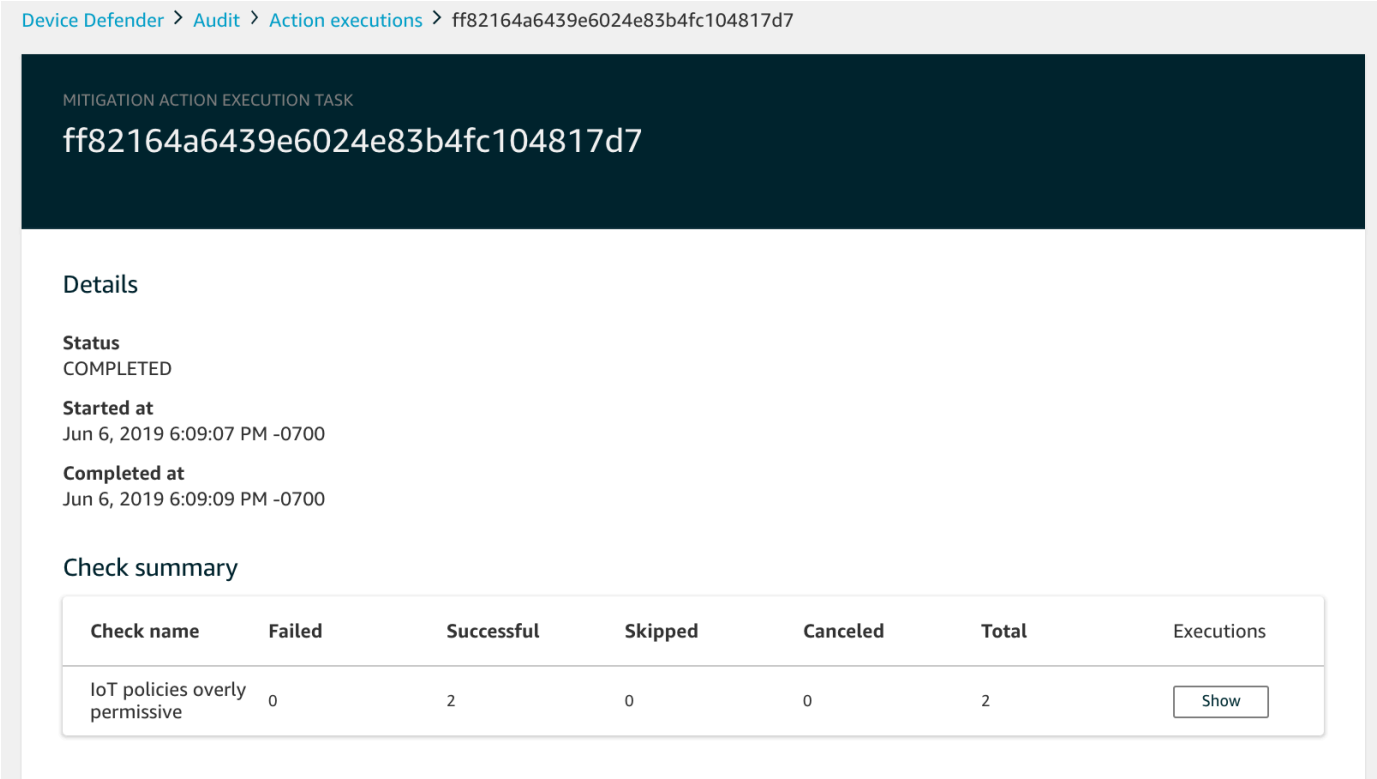
AWS CLI에서 감사 완화 작업 실행을 시작하여 완화 정책을 적용하려면

1. 작업을 모든 감사 결과에 적용하려면 ListAuditTasks 명령을 사용하여 작업 ID를 찾습니다.
2. 작업을 선택한 결과에만 적용하려면 ListAuditFindings 명령을 사용하여 결과 ID를 가져옵니다.
3. ListMitigationActions 명령을 사용하고 적용하려는 완화 작업의 이름을 기록해 둡니다.
4. StartAuditMitigationActionsTask 명령을 사용하여 작업을 대상에 적용할 수 있습니다. 작업 ID을 기록해 둡니다. ID를 사용하여 작업 실행 상태를 확인하거나 세부 정보를 검토하거나 취소할 수 있습니다.

AWS IoT 콘솔을 사용하여 작업 실행을 확인하려면

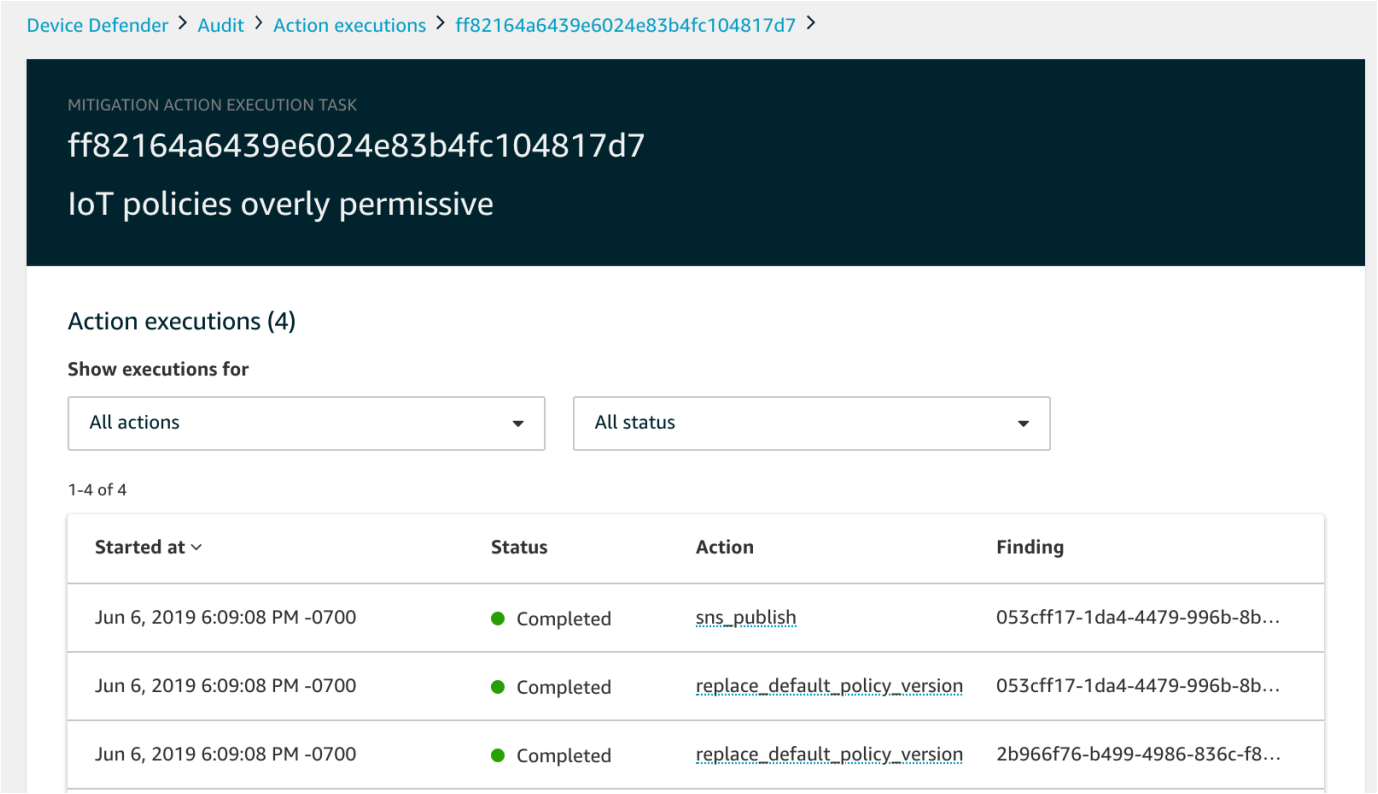
1. AWS IoT 콘솔의 Action tasks(작업 태스크) 페이지를 엽니다.

   작업 목록에는 각 작업이 시작된 시기와 현재 상태가 표시됩니다.
2. 작업의 세부 정보를 보려면 이름 링크를 선택합니다. 세부 정보에는 작업, 대상, 상태별로 적용된 작업이 모두 포함됩니다.



Show executions for(다음에 대한 실행 표시) 필터를 사용하여 작업 유형 또는 작업 상태에 초점을 맞출 수 있습니다.

3. 작업에 대한 세부 정보를 보려면 실행에서 표시를 선택합니다.



AWS CLI를 사용하여 시작한 작업의 목록을 생성하려면

1. ListAuditMitigationActionsTasks를 사용하여 감사 완화 작업 태스크를 봅니다. 결과를 줄이기 위한 필터를 제공할 수 있습니다. 작업의 세부 정보를 보려면 작업 ID를 기록해 둡니다.
2. ListAuditMitigationActionsExecutions를 사용하여 특정 감사 완화 작업의 실행 세부 정보를 확인합니다.
3. DescribeAuditMitigationActionsTask를 사용하여 시작 시 지정된 파라미터 등의 작업 관련 세부 정보를 확인합니다.

AWS CLI를 사용하여 실행 중인 감사 완화 작업을 취소하려면

1. ListAuditMitigationActionsTasks 명령을 사용하여 실행을 취소할 작업의 작업 ID를 찾습니다. 결과를 줄이기 위한 필터를 제공할 수 있습니다.
2. ListDetectMitigationActionsExecutions 명령으로 작업 ID를 사용하여 감사 완화 작업을 취소합니다. 완료된 작업은 취소할 수 없습니다. 작업을 취소하면 남은 작업은 적용되지 않지만 이미 적용된 완화 작업은 롤백되지 않습니다.

권한

정의한 각 완화 작업에 해당 작업을 적용하는 데 사용한 역할을 제공해야 합니다.

완화 작업 권한

작업 유형	권한 정책 템플릿
UPDATE_DEVICE_CERTIFICATE	`{ "Version":"2012-10-17", "Statement":[ { "Effect": "Allow", "Action":[ "iot:UpdateCertificate" ], "Resource": [ "*" ] } ] }`
UPDATE_CA_CERTIFICATE	`{ "Version":"2012-10-17", "Statement":[ { "Effect": "Allow", "Action":[ "iot:UpdateCACertificate"`

작업 유형	권한 정책 템플릿	

	``
	],
	"Resource":
	[
	"*"
	]
	}
	]
	}
	``	

ADD_THINGS_TO_THING_GROUP	```
	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":
	"Allow",
	"Action":[
	"iot:ListPrincipalThings",
	"iot:AddThingToThingGroup"
	],
	"Resource":
	[
	"*"
	]
	}
	]
	}
	``	

작업 유형	권한 정책 템플릿	
REPLACE_DEFAULT_POLICY_VERSION	`{ "Version":"2012-10-17", "Statement":[ { "Effect": "Allow", "Action":[ "iot:CreatePolicyVersion" ], "Resource": [ "*" ] } ] }`	

작업 유형	권한 정책 템플릿	
ENABLE_IOT_LOGGING	`{ "Version":"2012-10-17", "Statement":[ { "Effect": "Allow", "Action":[ "iot:SetV2LoggingOptions" ], "Resource": [ "*" ] }, { "Effect": "Allow", "Action":[ "iam:PassRole" ], "Resource": [ "<IAM role ARN used for setting up logging>" ] } ] }`	

작업 유형	권한 정책 템플릿	
PUBLISH_FINDING_TO_SNS	`{ "Version":"2012-10-17", "Statement":[ { "Effect": "Allow", "Action":[ "sns:Publish" ], "Resource": [ "<The SNS topic to which the finding is published> " ] } ] }`	

모든 완화 작업에 다음 신뢰 정책 템플릿을 사용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iot:*:111122223333::*"
        },
        "StringEquals": {
```

```
            "aws:SourceAccount": "111122223333:"
          }
        }
      }
    ]
}
```

완화 작업 명령

나중에 하나 이상의 감사 결과 집합에 적용할 수 있도록 AWS 계정에 대한 작업 집합을 정의하려면 완화 작업 명령을 사용할 수 있습니다. 명령 카테고리는 세 가지가 있습니다.

- 작업을 정의하고 관리하는 데 사용되는 명령.
- 감사 결과에 대한 작업 적용을 시작하고 관리하는 데 사용되는 명령.
- 감지 경보에 대한 작업 적용을 시작하고 관리하는 데 사용되는 명령.

완화 작업 명령

작업 정의 및 관리	감사 실행 시작 및 관리	감지(Detect) 실행 시작 및 관리
CreateMitigationAction	CancelAuditMitigationActionsTask	CancelDetectMitigationActionsTask
DeleteMitigationAction	DescribeAuditMitigationActionsTask	DescribeDetectMitigationActionsTask
DescribeMitigationAction	ListAuditMitigationActionsTasks	ListDetectMitigationActionsTasks
ListMitigationActions	StartAuditMitigationActionsTask	StartDetectMitigationActionsTask
UpdateMitigationAction	ListAuditMitigationActionsExecutions	ListDetectMitigationActionsExecutions

다른 AWS 서비스와 함께 AWS IoT Device Defender 사용

AWS IoT Greengrass을(를) 실행하는 디바이스와 함께 AWS IoT Device Defender 사용

AWS IoT Greengrass은(는) AWS IoT Device Defender와의 미리 빌드된 통합을 제공해 지속적으로 디바이스 동작을 모니터링합니다.

- AWS IoT Greengrass V1과 Device Defender 통합
- AWS IoT Greengrass V2와 Device Defender 통합

FreeRTOS 및 임베디드 디바이스와 함께 AWS IoT Device Defender 사용

FreeRTOS 디바이스에서 AWS IoT Device Defender를 사용하려면 디바이스에 FreeRTOS Embedded C SDK 또는 AWS IoT Device Defender 라이브러리가 설치되어 있어야 합니다. FreeRTOS Embedded C SDK에는 AWS IoT Device Defender 라이브러리가 포함되어 있습니다. AWS IoT Device Defender을(를) FreeRTOS와 통합하는 방법에 대한 자세한 내용은 다음 데모를 참조하세요.

- FreeRTOS용 AWS IoT Device Defender 표준 지표 및 사용자 정의 지표 데모
- MQTT 에이전트를 사용하여 AWS IoT Device Defender에 지표 제출
- MQTT 코어 라이브러리를 사용하여 AWS IoT Device Defender에 지표 제출

FreeRTOS 없이 임베디드 디바이스에서 AWS IoT Device Defender를 사용하려면 디바이스에 AWS IoT Embedded C SDK 또는 AWS IoT Device Defender 라이브러리가 있어야 합니다. AWS IoT Embedded C SDK에는 AWS IoT Device Defender 라이브러리가 포함되어 있습니다. AWS IoT Device Defender를 임베디드 디바이스와 통합하는 방법에 대한 자세한 내용은 다음 데모인 AWS IoT Device Defender for AWS IoT Embedded SDK 표준 및 사용자 정의 지표 데모를 참조하세요.

AWS IoT Device Management와 함께 AWS IoT Device Defender 사용

AWS IoT Device Management 플릿 인덱싱을 사용하여 AWS IoT Device Defender 탐지 위반을 인덱싱, 검색 및 집계할 수 있습니다. Device Defender 위반 데이터가 플릿 인덱싱에서 인덱싱된 후 Fleet Hub 애플리케이션에서 Device Defender 위반 데이터에 액세스 및 쿼리하고 위반 데이터를 기반으로 플릿 경보를 생성하여 디바이스 플릿 전반의 이상을 모니터링하고 Fleet Hub 대시보드에서 플릿 경보를 볼 수 있습니다.

> **Note**
>
> AWS IoT Device Defender 위반 데이터 인덱싱을 지원하는 플릿 인덱싱 기능은 AWS IoT Device Management의 미리 보기 릴리스에 있으며 변경될 수 있습니다.

- 플릿 인덱싱 관리
- 쿼리 구문
- Fleet Hub 애플리케이션에 대한 플릿 인덱싱 관리
- 시작하기

AWS Security Hub와의 통합

AWS Security Hub에서는 AWS에서 보안 상태를 포괄적으로 파악할 수 있으며 보안 업계 표준 및 모범 사례와 비교하여 환경을 확인할 수 있습니다. Security Hub는 AWS 계정, 서비스 및 지원되는 서드 파티 파트너 제품에서 보안 데이터를 수집합니다. Security Hub를 사용하여 보안 추세를 분석하고 우선순위가 가장 높은 보안 문제를 식별할 수 있습니다.

Security Hub와 AWS IoT Device Defender의 통합을 통해 AWS IoT Device Defender에서 Security Hub로 결과를 보낼 수 있습니다. 그러면 Security Hub의 보안 태세 분석에 이러한 결과가 포함됩니다.

목차

- 통합 활성화 및 구성
- AWS IoT Device Defender에서 Security Hub로 결과를 보내는 방법
  - AWS IoT Device Defender이(가) 보내는 결과의 유형
  - 결과 전송 지연 시간

- Security Hub 사용할 수 없을 때 다시 시도
- Security Hub에서 기존 결과 업데이트
- AWS IoT Device Defender의 일반적 결과
- AWS IoT Device Defender에서 결과를 Security Hub로 전송하는 작업 중지

통합 활성화 및 구성

AWS IoT Device Defender를 Security Hub와 통합하기 전에 먼저 Security Hub를 활성화해야 합니다. Security Hub를 활성화하는 방법에 대한 자세한 내용은 AWS Security Hub 사용 설명서의 Security Hub 설정을 참조하세요.

AWS IoT Device Defender와 Security Hub를 모두 활성화한 후 Security Hub 콘솔의 Integrations(통합) 페이지를 열고 Audit, Detect 또는 둘 다에 대해 Accept findings(분석 결과 수락)를 선택합니다. AWS IoT Device Defender가 결과를 Security Hub로 보내기 시작합니다.

AWS IoT Device Defender에서 Security Hub로 결과를 보내는 방법

Security Hub의 경우 보안 문제를 결과와 같이 추적합니다. 일부 결과는 다른 AWS 서비스 또는 서드 파티 제품에서 감지한 문제에서 비롯됩니다.

Security Hub는 이러한 모든 출처를 총망라하여 결과를 관리할 도구를 제공합니다. 사용자는 결과 목록을 조회하고 필터링할 수 있으며 주어진 결과의 세부 정보를 조회할 수도 있습니다. 자세한 내용은 AWS Security Hub User Guide의 Viewing findings를 참조하세요. 또한 주어진 결과에 대한 조사 상태를 추적할 수도 있습니다. 자세한 내용은 AWS Security Hub User Guide의 Taking action on findings를 참조하세요.

Security Hub의 모든 결과는 표준 JSON 형식을 사용합니다. 이를 AWS Security Finding Format(ASFF)이라고 합니다. ASFF에는 문제의 출처, 영향을 받은 리소스와 결과의 현재 상태 등에 관한 세부 정보가 포함됩니다. 자세한 내용은 AWS Security Hub 사용 설명서의 AWS Security Finding Format(ASFF)을 참조하세요.

AWS IoT Device Defender는 Security Hub에 결과를 전송하는 AWS 서비스 중 하나입니다.

AWS IoT Device Defender이(가) 보내는 결과의 유형

Security Hub와의 통합을 활성화한 후 AWS IoT Device Defender Audit는 생성된 결과(검사 요약이라고 함)를 Security Hub로 보냅니다. 검사 요약은 특정 감사 검사 유형 및 특정 감사 작업에 대한 일반적인 정보입니다. 자세한 내용은 감사 검사를 참조하세요.

AWS IoT Device Defender Audit는 각 감사 작업의 감사 검사 요약과 감사 결과 모두에 대한 결과 업데이트를 Security Hub로 보냅니다. 감사 검사에서 찾은 모든 리소스가 규정을 준수하거나 감사 작업이 취소된 경우 Audit는 Security Hub의 검사 요약을 ARCHIVED 레코드 상태로 업데이트합니다. 리소스가 감사 검사에서 규정 미준수로 보고되었지만 마지막 감사 작업에서 규정 준수로 보고된 경우 Audit는 리소스를 규정 준수로 변경하고 Security Hub의 결과를 ARCHIVED 레코드 상태로 업데이트합니다.

AWS IoT Device Defender Detect는 Security Hub로 위반 결과를 전송합니다. 이러한 위반 결과에는 기계 학습(ML), 통계 및 정적 동작이 포함됩니다.

결과를 Security Hub로 보내기 위해 AWS IoT Device Defender는 AWS Security Finding Format(ASFF)을 사용합니다. ASFF의 경우, `Types` 필드가 결과 유형을 제공합니다. AWS IoT Device Defender의 결과는 `Types`의 값이 다음과 같을 수 있습니다.

비정상 동작

충돌하는 MQTT 클라이언트 ID 및 공유된 디바이스 인증서 점검의 결과 유형과 Detect의 결과 유형입니다.

소프트웨어와 구성 점검 및 취약성

기타 모든 감사 검사의 결과 유형입니다.

결과 전송 지연 시간

AWS IoT Device Defender Audit에서 새 결과를 생성하면 감사 작업이 완료된 후 즉시 Security Hub로 전송됩니다. 지연 시간은 감사 작업에서 생성된 결과의 양에 따라 달라집니다. Security Hub 허브는 일반적으로 1시간 이내에 결과를 받습니다.

AWS IoT Device Defender Detect는 거의 실시간으로 Security 결과를 전송합니다. 위반이 경보에 포함되거나 해제된 후(경보가 생성 또는 삭제됨을 의미) 해당 Security Hub 결과가 즉시 생성되거나 보관됩니다.

Security Hub 사용할 수 없을 때 다시 시도

Security Hub를 사용할 수 없는 경우 AWS IoT Device Defender Audit 및 AWS IoT Device Defender Detect는 결과가 수신될 때까지 결과 전송을 재시도합니다.

Security Hub에서 기존 결과 업데이트

AWS IoT Device Defender Audit 결과가 Security Hub로 전송된 후에는 검사된 리소스 식별자 및 감사 검사 유형으로 이를 식별할 수 있습니다. 동일한 리소스 및 감사 점검에 대한 후속 감사 작업으로 새 감

사 결과가 생성되면 AWS IoT Device Defender Audit는 결과 활동의 추가 관찰 결과를 반영한 업데이트를 Security Hub로 보냅니다. 동일한 리소스 및 감사 검사에 대한 후속 감사 작업에서 추가 감사 결과가 생성되지 않으면 리소스가 감사 검사를 준수하는 것으로 변경됩니다. AWS IoT Device Defender Audit는 그 결과를 Security Hub에 보관합니다.

AWS IoT Device Defender Audit는 Security Hub의 검사 요약도 업데이트합니다. 감사 검사에서 미준수 리소스가 발견되거나 검사가 실패하는 경우 Security Hub 결과가 활성 상태가 됩니다. 그렇지 않은 경우 AWS IoT Device Defender Audit는 그 결과를 Security Hub에 보관합니다.

AWS IoT Device Defender Detect는 위반(예: 경보)이 있을 때 Security Hub 결과를 생성합니다. 이 결과는 다음 기준 중 하나를 충족할 때만 업데이트됩니다.

- 결과가 Security Hub에서 곧 만료되어 AWS IoT Device Defender에서 결과를 최신 상태로 유지하기 위해 업데이트를 보냅니다. 결과는 가장 최근 업데이트 후 90일 또는 업데이트가 없는 경우 생성일 이후 90일에 삭제됩니다. 자세한 내용은 AWS Security Hub 사용 설명서의 Security Hub 할당량을 참조하세요.
- 해당 위반에 대한 경보가 해제되어 AWS IoT Device Defender에서 결과 상태를 ARCHIVED로 업데이트합니다.

AWS IoT Device Defender의 일반적 결과

AWS IoT Device Defender는 AWS Security Finding Format(ASFF)을 사용하여 결과를 Security Hub로 보냅니다.

다음 예는 감사 결과에 대한 Security Hub의 일반적인 결과를 보여줍니다. `ProductFields`의 `ReportType`은 `AuditFinding`입니다.

```
  {
  "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"
    ]
  }
}
```

다음 예는 감사 검사 요약에 대한 Security Hub의 결과를 보여줍니다. `ProductFields`의 `ReportType`은 `CheckSummary`입니다.

```
  {
  "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"
    ]
  }
}
```

다음 예는 AWS IoT Device Defender Detect 위반에 대한 Security Hub의 일반적인 결과를 보여줍니다.


```
  {
  "SchemaVersion": "2018-10-08",
  "Id": "e92a782593c6f5b1fc7cb6a443dc1a12",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/iot-device-defender-
detect",
  "ProductName": "IoT Device Defender - Detect",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "arn:aws:iot:us-east-1:123456789012:securityprofile/
MySecurityProfile",
  "AwsAccountId": "123456789012",
  "Types": [
    "Unusual Behaviors"
  ],
  "CreatedAt": "2022-11-09T22:45:00Z",
  "UpdatedAt": "2022-11-09T22:45:00Z",
  "Severity": {
    "Label": "MEDIUM",
    "Normalized": 40
  },
  "Title": "Registered thing MyThing is in alarm for STATIC behavior MyBehavior.",
  "Description": "Registered thing MyThing violates STATIC behavior MyBehavior of
 security profile MySecurityProfile. Violation was triggered because the device did not
 conform to aws:num-disconnects less-than 1.",
  "SourceUrl": "https://us-east-1.console.aws.amazon.com/iot/home?region=us-east-1#/dd/
securityProfile/MySecurityProfile?tab=violations",
  "ProductFields": {
    "ComparisonOperator": "less-than",
    "BehaviorName": "MyBehavior",
    "ViolationId": "e92a782593c6f5b1fc7cb6a443dc1a12",
    "ViolationStartTime": "1668033900000",
    "SuppressAlerts": "false",
    "ConsecutiveDatapointsToAlarm": "1",
    "ConsecutiveDatapointsToClear": "1",
    "DurationSeconds": "300",
    "Count": "1",
    "MetricName": "aws:num-disconnects",
```

```
    "BehaviorCriteriaType": "STATIC",
    "ThingName": "MyThing",
    "SecurityProfileName": "MySecurityProfile",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/iot-
device-defender-detect/e92a782593c6f5b1fc7cb6a443dc1a12",
    "aws/securityhub/ProductName": "IoT Device Defender - Detect",
    "aws/securityhub/CompanyName": "AWS"
  },
  "Resources": [
    {
      "Type": "AwsIotRegisteredThing",
      "Id": "MyThing",
      "Region": "us-east-1",
      "Details": {
        "Other": {
          "SourceUrl": "https://us-east-1.console.aws.amazon.com/iot/home?region=us-
east-1#/thing/MyThing?tab=violations",
          "IsRegisteredThing": "true",
          "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyThing"
        }
      }
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "MEDIUM"
    },
    "Types": [
      "Unusual Behaviors"
    ]
  }
}
```

AWS IoT Device Defender에서 결과를 Security Hub로 전송하는 작업 중지

Security Hub로 결과를 전송하는 작업을 중지하려면 Security Hub 콘솔 또는 API를 사용하면 됩니다.

자세한 내용은 AWS Security Hub 사용 설명서에서 통합에서 결과 흐름 활성화 및 비활성화(콘솔) 또는 통합에서 결과 흐름 비활성화(Security Hub API, AWS CLI)를 참조하세요.

교차 서비스 혼동된 대리인 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. AWS에서는 교차 서비스 가장으로 인해 혼동된 대리자 문제가 발생할 수 있습니다. 교차 서비스 가장은 한 서비스(호출하는 서비스)가 다른 서비스(호출되는 서비스)를 직접적으로 호출할 때 발생할 수 있습니다. 호출하는 서비스는 호출된 서비스를 통해 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다.

사용자로부터 AWS IoT Device Defender가 액세스하는 세 가지 리소스는 혼동을 유발하는 대리 보안 문제, 감사 실행, 보안 프로필 위반에 대한 SNS 알림 전송 및 완화 작업 실행으로 인해 영향을 받을 수 있습니다. 이러한 각 작업에 대해 `aws:SourceArn`의 값은 다음과 같아야 합니다.

- UpdateAccountAuditConfiguration API(RoleArn 및 notificationTarget RoleArn 속성)에서 전달된 리소스의 경우 `aws:SourceArn`을 `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:`로 사용하여 리소스 정책의 범위를 좁혀야 합니다.
- CreateMitigationAction API(RoleArn 속성)에서 전달된 리소스의 경우 `aws:SourceArn`을 `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:mitigationaction/`*`mitigationActionName`*으로 사용하여 리소스 정책의 범위를 좁혀야 합니다.
- CreateSecurityProfile API(alertTargets 속성)에서 전달된 리소스의 경우 `aws:SourceArn`을 `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:securityprofile/`*`securityprofileName`*으로 사용하여 리소스 정책의 범위를 좁혀야 합니다.

혼동된 대리자 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 ARN이 포함된 `aws:SourceArn` 전역 조건 컨텍스트 키를 사용하는 것입니다. 리소스의 전체 ARN을 모를 경우 또는 여러 리소스를 지정하는 경우, ARN의 알 수 없는 부분에 대해 와일드카드(*)를 포함한 `aws:SourceArn` 전역 조건 컨텍스트 키를 사용합니다. 예제: `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

다음 예는 AWS IoT Device Defender에서 `aws:SourceArn` 및 `aws:SourceAccount` 전역 조건 컨텍스트 키를 사용하여 혼동된 대리자 문제를 방지하는 방법을 보여줍니다.

```
 {
"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:"
    }
  }
}
}
```

디바이스 에이전트의 보안 모범 사례

최소 권한

에이전트 프로세스는 이 업무를 수행하는 데 필요한 최소 권한만 부여해야 합니다.

기본 메커니즘

- 에이전트를 루트 이외 사용자로 실행해야 합니다.
- 에이전트는 자체 그룹에서 전용 사용자로 실행해야 합니다.
- 사용자/그룹에는 지표를 수집하고 전송하는 데 필요한 리소스에 대한 읽기 전용 권한만 부여해야 합니다.
- 예: 샘플 에이전트의 /proc /sys에 대한 읽기 전용
- 프로세스가 감소된 권한으로 실행되도록 설정하는 방법에 대한 예는 Python 샘플 에이전트에 포함된 설정 지침을 참조하세요.

에이전트 프로세스를 더 제한 또는 격리하는 데 도움이 되는 잘 알려진 여러 가지 Linux 메커니즘이 있습니다.

고급 메커니즘

- CGroups

- SELinux
- Chroot
- Linux 네임스페이스

운영 복원성

에이전트 프로세스는 예상치 못한 운영 오류 및 예외에 대한 복원력이 있어야 하며, 영구적으로 종료되거나 충돌해서는 안 됩니다. 코드는 예외를 정상적으로 처리해야 하며, 만일을 위해 예상치 못한 종료가 발생할 경우(예: 시스템 다시 시작 또는 확인할 수 없는 예외로 인해) 자동으로 다시 시작하도록 구성되어야 합니다.

최소 종속성

에이전트는 구현 시 가능한 최소 수의 종속성(즉, 타사 라이브러리)을 사용해야 합니다. 작업의 복잡성(예: 전송 계층 보안)으로 인해 라이브러리의 사용이 정당한 경우 잘 유지된 종속성만 사용하고 종속성을 최신 상태로 유지하는 메커니즘을 설정합니다. 에이전트가 사용하지 않으며 기본적으로 활성 상태인 기능(예: 개방형 포트, 도메인 소켓)이 포함된 종속성이 추가된 경우 코드에서 비활성화하거나 라이브러리의 구성 파일로 비활성화합니다.

프로세스 격리

에이전트 프로세스에는 디바이스 지표 수집과 전송을 수행하는 데 필요한 기능만 포함되어야 합니다. 다른 시스템의 프로세스를 컨테이너로 피기 백하거나 범위를 벗어난 기타 사용 사례에 기능을 구현해서는 안 됩니다. 또한, 에이전트 프로세스에서는 로컬 또는 원격 프로세스가 작업을 방해하며 무결성 및 격리에 영향을 줄 수 있는 도메인 소켓과 네트워크 서비스 포트와 같은 인바운드 통신 채널을 생성하지 않아야 합니다.

은밀함

보안, 모니터링, 감사와 같이 목적 및 보안 값을 나타내는 키워드로 에이전트 프로세스의 이름을 지정해서는 안 됩니다. 일반 코드 이름 또는 디바이스마다 무작위의 고유한 프로세스 이름이 선호됩니다. 에이전트의 바이너리가 있는 디렉터리의 이름 지정과 프로세스 인수의 모든 이름 및 값 지정과 동일한 원칙을 따라야 합니다.

최소 정보 공유

디바이스에 배포된 에이전트 아티팩트에는 권한 있는 자격 증명, 디버깅과 데드 코드 또는 에이전트가 수집한 지표의 서버 측 처리에 대한 세부 정보 또는 백엔드 시스템에 대한 기타 세부 정보를 드러내는 인라인 설명이나 설명서 파일 등 중요한 정보가 없어야 합니다.

전송 계층 보안

데이터 전송을 위한 TLS 보안 채널을 설정하기 위해 에이전트 프로세스에서는 기본적으로 활성화되지 않은 경우 애플리케이션 수준에서 인증서 체인 및 도메인 이름 검증과 같은 모든 클라이언트

측 검증을 적용해야 합니다. 또한 에이전트는 신뢰할 수 있는 기관을 포함하지만 손상된 인증서 발급자에 속한 인증서를 포함하지 않는 루트 인증서 스토어를 사용해야 합니다.

안전한 배포

코드 푸시 또는 동기화와 같은 에이전트 배포 메커니즘과 바이너리, 소스 코드 및 모든 구성 파일(신뢰할 수 있는 루트 인증서 포함)이 포함된 리포지토리에는 무단 코드 주입이나 훼손을 방지하기 위해 액세스가 제어되어야 합니다. 배포 메커니즘이 네트워크 통신을 토대로 하는 경우 전송 중인 배포 아티팩트의 무결성을 보호하기 위해 암호화 방식을 사용합니다.

참고 문헌

- AWS IoT Device Defender의 보안
- AWS IoT 보안 모델에 대한 이해
- Redhat: Python의 일부
- Python의 10가지 공통 보안 실수 및 이를 방지하는 방법
- 최소 권한이란 무엇이며 최소 권한이 필요한 이유는 무엇입니까?
- OWASP의 10대 임베디드 보안
- OWASP IoT 프로젝트

AWS IoT Device Defender 문제 해결 안내서

이 주제를 개선하도록 도와주세요.

이 주제를 개선할 수 있는 의견이 있으시다면 알려주세요.

일반

Q: AWS IoT Device Defender 사용에 대한 사전 조건이 있습니까?

A: 디바이스 보고 지표를 사용하려면 먼저 AWS IoT가 연결된 디바이스 또는 디바이스 게이트웨이에 에이전트를 배포해야 합니다. 디바이스는 일관된 클라이언트 식별자 또는 사물 이름을 제공해야 합니다.

감사

Q: 점검을 활성화했으며 내 감사에 장시간 "진행 중"이 표시되고 있습니다. 무언가 잘못되었습니까? 언제 결과가 나옵니까?

A: 점검이 활성화되었을 때 데이터 수집이 즉시 시작됩니다. 그러나, 계정에서 수집할 데이터(예: 인증서, 사물 또는 정책)의 양이 많은 경우 점검을 활성화한 후 점검 결과를 사용할 수 없는 경우가 있을 수 있습니다.

감지

Q: AWS IoT Device Defender 보안 프로필 동작에 설정되어야 하는 임계값을 어떻게 알 수 있습니까?

A: 먼저 낮은 임계값으로 보안 프로필 동작을 생성하고 이 동작을 대표적인 디바이스로 구성된 사물 그룹에 연결합니다. AWS IoT Device Defender를 사용하여 현재 지표를 본 다음 사용 사례에 맞춰 디바이스 동작 임계값을 미세 조정할 수 있습니다.

Q: 동작을 생성했지만, 예상과 달리 위반을 트리거하지 않습니다. 어떻게 해결해야 합니까?

A: 동작을 정의할 때 디바이스가 정상적으로 동작하는 방법을 지정해야 합니다. 예를 들어, 보안 카메라가 TCP 포트 8888에 있는 중앙 서버 하나에만 연결되는 경우 다른 연결이 수행될 것으로 기대되지 않습니다. 카메라가 다른 포트에서 연결될 경우 알림을 받기 위해 다음과 같이 동작을 정의합니다.

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 8888 ]
    }
  }
}
```

카메라가 TCP 포트 443에 대해 TCP 연결을 수행하는 경우, 디바이스 동작이 위반되며 알림이 트리거됩니다.

Q: 하나 이상의 동작이 위반되었습니다. 위반을 어떻게 지울 수 있습니까?

A: 동작 프로필에서 정의된 대로 디바이스가 예상된 동작을 반환하면 경보가 지워집니다. 디바이스에 대한 지표 데이터를 받을 때 동작 프로필이 평가됩니다. 디바이스가 2일 이상 지표를 게시하지 않으면 위반 이벤트가 자동으로 `alarm-invalidated`로 설정됩니다.

Q: 위반된 동작을 삭제했습니다만 알림을 어떻게 중지할 수 있습니까?

A: 동작을 삭제하면 해당 동작에 대한 향후 위반 및 알림이 모두 중지됩니다. 이전 알림은 알림 메커니즘에서 드레이닝해야 합니다. 동작을 삭제하면 해당 동작의 위반 기록이 계정의 다른 모든 위반과 동일한 기간 동안 유지됩니다.

디바이스 지표

Q: 내 동작을 위반한 지표 보고서를 제출했지만 위반이 트리거되지 않습니다. 무엇이 문제인가요?

A: 다음 MQTT 주제를 구독하여 지표 보고서가 수락되었는지 확인하세요.

```
$aws/things/THING_NAME/defender/metrics/FORMAT/rejected
$aws/things/THING_NAME/defender/metrics/FORMAT/accepted
```

`THING_NAME`은 지표를 보고하는 사물의 이름이며, `FORMAT`은 사물이 제출하는 지표 보고서의 형식에 따라 "JSON" 또는 "CBOR"입니다.

구독하면 제출한 각 지표 보고서의 주제에 대한 메시지를 받아야 합니다. `rejected` 메시지는 지표 보고서를 구분 분석하는 동안 문제가 있음을 나타냅니다. 지표 보고서의 오류를 해결하는 데 도움이 되도록 메시지 페이로드에 오류 메시지가 포함됩니다. `accepted` 메시지는 지표 보고서가 제대로 구문 분석되었음을 나타냅니다.

Q: 내 지표 보고서에 빈 지표를 전송하면 어떻게 됩니까?

A: 포트 또는 IP 주소의 비어 있는 목록은 항상 해당 동작을 따르고 있다고 간주됩니다. 해당 동작이 위반되는 경우 위반이 지워집니다.

Q: 내 디바이스 지표 보고서에 AWS IoT 레지스트리에 없는 디바이스에 대한 메시지가 포함된 이유는 무엇입니까?

하나 이상의 보안 프로필이 모든 사물 또는 등록되지 않은 모든 사물에 연결된 경우 AWS IoT Device Defender에는 등록되지 않은 사물의 지표가 포함됩니다. 등록되지 않은 사물에서 지표를 제외할 경우 모든 디바이스 대신 등록된 모든 디바이스에 프로필을 연결할 수 있습니다.

Q: 등록되지 않은 모든 디바이스 또는 모든 디바이스에 보안 프로필을 적용했는데도 하나 이상의 등록되지 않은 디바이스의 메시지가 보이지 않습니다. 해결하려면 어떻게 해야 합니까?

지원되는 형식 중 하나를 사용하여 올바른 지표 보고서를 제출하는지 확인하세요. 자세한 내용은 디바이스 지표 문서 사양 단원을 참조하십시오. 등록되지 않은 디바이스가 일관된 클라이언트 식별자 또는 사물 이름을 사용하고 있는지 확인하세요. 사물 이름에 제어 문자가 포함되어 있거나 사물 이름이 UTF-8로 인코딩된 문자 128바이트보다 긴 경우 디바이스에서 보고된 메시지는 거부됩니다.

Q: 등록되지 않은 디바이스가 레지스트리에 추가되거나 등록된 디바이스가 등록 취소되면 어떻게 됩니까?

A: 디바이스가 레지스트리에 추가되거나 레지스트리에서 제거되는 경우 다음과 같습니다.

- 디바이스에서 위반에 대한 지표를 계속 게시하는 경우 디바이스에 대한 두 개의 별도 위반(등록된 사물 이름으로 위반, 등록되지 않은 자격 증명으로 위반)이 표시됩니다. 이전 자격 증명에 대한 활성 위반은 2일 후에 나타나지 않지만 최대 14일 동안 위반 기록에서 확인할 수 있습니다.

Q: 내 디바이스 지표 보고서의 보고서 ID 필드에 어떤 값을 입력해야 합니까?

A: 각 지표 보고서에 대한 고유한 값을 사용하며 이 값은 양의 정수로 표현되어야 합니다. 일반적인 관행은 Unix epoch 타임스탬프를 사용하는 것입니다.

Q: AWS IoT Device Defender 지표에 대한 전용 MQTT 연결을 생성해야 합니까?

A: 별도의 MQTT 연결은 필요하지 않습니다.

Q: 디바이스 지표를 게시하기 위해 연결할 때 어떤 클라이언트 ID를 사용해야 합니까?

AWS IoT 레지스트리에 있는 디바이스(사물)의 경우 등록된 사물 이름을 사용합니다. AWS IoT 레지스트리에 없는 디바이스의 경우 AWS IoT에 연결할 때 일관된 식별자를 사용합니다. 이 관행은 위반을 사물 이름과 일치시키는 데 도움이 됩니다.

Q: 다른 클라이언트 ID의 디바이스에 대한 지표를 게시할 수 있습니까?

다른 사물을 대신하여 지표를 게시할 수 있습니다. 해당 디바이스에 대한 AWS IoT Device Defender 예약된 주제에 지표를 게시하면 됩니다. 예를 들어 `Thing-1`은 `Thing-2`를 대신하여 자체적으로 지표를 게시하려고 합니다. `Thing-1`은 자체 지표를 수집하여 MQTT 주제에 게시합니다.

```
$aws/things/Thing-1/defender/metrics/json
```

그런 다음 `Thing-1`은 `Thing-2`에서 지표를 가져오고 해당 지표를 MQTT 주제에 게시합니다.

```
$aws/things/Thing-2/defender/metrics/json
```

Q: 내 계정에 보안 프로필과 동작을 얼마나 보유할 수 있습니까?

A: AWS IoT Device Defender 엔드포인트 및 할당량을 참조하세요.

Q: 알림 대상에 대한 원형 대상 역할은 어떤 형태입니까?

A: AWS IoT Device Defender가 알림 대상(SNS 주제)에 알림을 게시할 수 있는 역할에는 다음과 같은 2개의 사물이 필요합니다.

- iot.amazonaws.com을 신뢰할 수 있는 개체로 지정하는 신뢰 관계
- 지정된 SNS 주제에 게시할 수 있는 AWS IoT 권한을 부여하는 연결된 정책 예:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "<sns-topic-arn>"
        }
    ]
}
```

- 알림 게시에 사용되는 SNS 주제가 암호화된 주제인 경우 SNS 주제 게시 권한과 함께 두 가지 추가 권한을 AWS IoT에 부여해야 합니다. 예:

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

```
            "Effect": "Allow",
            "Action": [
                "sns:Publish",
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "<sns-topic-arn>"
        }
    ]
}
```

Q: 사용자 정의 지표 유형 `number`가 포함된 내 지표 보고서 제출이 오류 메시지 `Malformed metrics report`와 함께 실패합니다. 무엇이 문제인가요?

A: `number` 유형은 단일 지표 값만 입력으로 사용하지만 DeviceMetrics 보고서에서 지표 값을 제출할 때에는 단일 값을 가진 배열로 전달해야 합니다. 지표 값을 배열로 제출하는지 확인합니다.

오류 페이로드:

```
{"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}}}
```

오류 메시지:

```
{"thingName":"myThing","status":"REJECTED","statusDetails":
{"ErrorCode":"InvalidPayload","ErrorMessage":"Malformed metrics
 report"},"timestamp":1635802047699}
```

오류 없는 페이로드:

```
{"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}]}}
```

응답:

```
{"thingName":"myThing","12334567":1635800375,"status":"ACCEPTED","timestamp":1635801636023}
```

AWS IoT Device Defender의 보안

AWS에서 클라우드 보안은 가장 중요합니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 고객의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안 - AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. AWS IoT Device Defender에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램의 범위에 속하는 AWS서비스 를 참조하세요.
- 클라우드의 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 AWS IoT Device Defender 사용 시 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 AWS IoT Device Defender을(를) 구성하는 방법을 보여줍니다. 또한 AWS IoT Device Defender 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스 사용 방법을 알아봅니다. AWS IoT Core의 보안에 대해 자세히 알아보려면 AWS IoT Core 개발자 안내서의 보안 장을 참조하세요.

주제

- AWS IoT Device Defender의 데이터 보호
- AWS IoT Device Defender의 I자격 증명 및 액세스 관리
- AWS IoT Device Defender의 규정 준수 검증
- AWS IoT Device Defender의 복원성

AWS IoT Device Defender의 데이터 보호

AWS 공동 책임 모델은 AWS IoT Device Defender의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시

FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정보안 인증 정보를 보호하고 AWS IAM Identity Center또는 AWS Identity and Access Management(IAM)를 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신하세요. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정하세요. AWS 활동 캡처에 CloudTrail 추적을 사용하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용하세요.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령행 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-3 검증된 암호화 모듈이 필요한 경우, FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 AWS IoT Device Defender 또는 기타 AWS 서비스 작업을 하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버로 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함해서는 안 됩니다.

AWS IoT Device Defender의 I자격 증명 및 액세스 관리

AWS Identity and Access Management(IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 통제할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 AWS IoT Device Defender 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 소유)를 받을 사용자를 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

주제

- 고객
- ID를 통한 인증
- 정책을 사용한 액세스 관리
- AWS IoT Device Defender에서 IAM을 사용하는 방식
- AWS IoT Device Defender에 대한 자격 증명 기반 정책 예시
- AWS IoT Device Defender 보안 인증 및 액세스 문제 해결

고객

AWS Identity and Access Management(IAM)을 사용하는 방법은 AWS IoT Device Defender에서 수행하는 작업에 따라 달라집니다.

서비스 사용자 – AWS IoT Device Defender 서비스를 사용하여 작업을 수행하는 경우 필요한 보안 인증 정보와 권한을 관리자가 제공합니다. 더 많은 AWS IoT Device Defender 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. AWS IoT Device Defender의 기능에 액세스할 수 없는 경우 AWS IoT Device Defender 보안 인증 및 액세스 문제 해결 단원을 참조하세요.

서비스 관리자 – 회사에서 AWS IoT Device Defender 리소스를 책임지고 있는 경우 AWS IoT Device Defender에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 AWS IoT RoboRunner 기능과 리소스를 결정합니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사가 AWS IoT Device Defender에서 IAM을 사용하는 방법에 대해 자세히 알아보려면 AWS IoT Device Defender에서 IAM을 사용하는 방식 단원을 참조하세요.

IAM 관리자 – IAM 관리자라면 AWS IoT Device Defender에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 AWS IoT Device Defender 자격 증명 기반 정책 예제를 보려면 AWS IoT Device Defender에 대한 자격 증명 기반 정책 예시 단원을 참조하세요.

ID를 통한 인증

인증은 ID 보안 인증을 사용하여 AWS에 로그인하는 방식입니다. AWS 계정 루트 사용자이나 IAM 사용자로, 또는 IAM 역할을 수임하여 인증(AWS에 로그인)받아야 합니다.

ID 소스를 통해 제공된 보안 인증 정보를 사용하여 페더레이션 ID로 AWS에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증, Google 또는

Facebook 보안 인증이 페더레이션 ID의 예시입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 연동을 사용하여 AWS에 액세스하면 간접적으로 역할을 수임합니다.

사용자 유형에 따라AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. AWS에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법을 참조하세요.

AWS에 프로그래밍 방식으로 액세스하는 경우, AWS에서는 보안 인증 정보를 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK) 및 명령줄 인터페이스(CLI)를 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 API 요청용 AWS Signature Version 4를 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS는 다중 인증(MFA)을 사용하여 계정의 보안을 강화하는 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 다중 인증 및 IAM 사용 설명서의 IAM의 AWS Multi-factor authentication을 참조하세요.

AWS 계정 루트 사용자

AWS 계정(을)를 생성할 때는 해당 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 단일 로그인 ID로 시작합니다. 이 ID는 AWS 계정루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 작업을 참조하십시오.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 보안 인증 정보를 사용하여 AWS 서비스에 액세스하도록 요구합니다.

페더레이션 ID는 엔터프라이즈 사용자 디렉터리, 웹 ID 공급자, AWS Directory Service, Identity Center 디렉터리의 사용자 또는 보안 인증 정보 소스를 통해 제공된 보안 인증 정보를 사용하여 AWS 서비스에 액세스하는 모든 사용자입니다. 페더레이션 ID는 AWS 계정에 액세스할 때 역할을 수임하고 역할은 임시 보안 인증 정보를 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 모든 AWS 계정 및 애플리케이션에서 사용하기 위해 고유한

ID 소스의 사용자 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇인가요?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가지고 있는 AWS 계정내 ID입니다. 가능하면 암호 및 액세스 키와 같은 장기 보안 인증이 있는 IAM 사용자를 생성하는 대신 임시 보안 인증을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 보안 인증이 필요한 특정 사용 사례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체를 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세히 알아보려면 IAM 사용자 가이드의 IAM 사용자 사용 사례를 참조하세요.

IAM 역할

IAM 역할은 특정 권한을 가지고 있는 AWS 계정 계정 내 ID입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. AWS Management Console에서 일시적으로 IAM 역할을 맡으려면 사용자에서 IAM 역할로 전환(콘솔)하면 됩니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 관련 역할에 대한 자세한 내용은 IAM 사용 설명서의 Create a role for a third-party identity provider (federation)를 참조하세요. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 작업에 대한 다양한 권한을 임시로 받을 수 있습니다.

- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스를 사용하면 (역할을 프록시로 사용하는 대신) 리소스에 정책을 직접 연결할 수 있습니다. 크로스 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.
- 교차 서비스 액세스 - 일부 AWS 서비스는 다른 AWS 서비스의 기능을 사용합니다. 예를 들어 서비스에서 호출하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 사람은 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 AWS 서비스를 직접 호출하는 보안 주체의 권한과 요청하는 AWS 서비스를 함께 사용하여 다운스트림 서비스에 대한 요청을 수행합니다. FAS 요청은 서비스에서 완료를 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 받은 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 Create a role to delegate permissions to an AWS 서비스를 참조하세요.
  - 서비스 연결 역할 – 서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 유형입니다. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 링크 역할은 AWS 계정에 나타나고, 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수는 없습니다.
- Amazon EC2에서 실행 중인 애플리케이션 – IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 보안 인증을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS역할을 할당하고 해당 역할을 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

정책을 사용한 액세스 관리

정책을 생성하고 AWS ID 또는 리소스에 연결하여 AWS에서 내 액세스를 제어합니다. 정책은 ID 또는 리소스와 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. AWS는 보안 주체(사용자, 루트 사용자

또는 역할 세션)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은 AWS에 JSON 문서로 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

관리자는 AWSJSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console, AWS CLI또는 AWSAPI에서 역할 정보를 가져올 수 있습니다.

보안 인증 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 만드는 방법을 알아보려면 IAM 사용자 가이드의 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 AWS 계정에 속한 다수의 사용자, 그룹 및 역할에 독립적으로 추가할 수 있는 정책입니다. 관리형 정책에는 AWS관리형 정책과 고객 관리형 정책이 포함되어 있습니다. 관리형 정책 또는 인라인 정책 간에 선택하는 방법을 알아보려면 IAM 사용자 가이드의 관리형 정책 및 인라인 정책 간에 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예시는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 AWS 서비스가 포함될 수 있습니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3, AWS WAF 및 Amazon VPC는 ACL을 지원하는 대표적인 서비스입니다. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 액세스 제어 목록(ACL) 개요를 참조하세요.

기타 정책 타입

AWS는 비교적 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 자격 증명 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 객체의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계를 참조하세요.
- 서비스 제어 정책(SCP) – SCP는 AWS Organizations에서 조직 또는 조직 단위(OU)에 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations는 기업이 소유하는 여러 개의 AWS 계정을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각 AWS 계정 루트 사용자을 비롯하여 멤버 계정의 엔터티에 대한 권한을 제한합니다. 조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서에서 Service control policies을 참조하세요.
- 리소스 제어 정책(RCP) - RCP는 소유한 각 리소스에 연결된 IAM 정책을 업데이트하지 않고 계정의 리소스에 대해 사용 가능한 최대 권한을 설정하는 데 사용할 수 있는 JSON 정책입니다. RCP는 멤버 계정의 리소스에 대한 권한을 제한하며 조직에 속하는지 여부에 관계없이 AWS 계정 루트 사용자를 포함한 ID에 대한 유효 권한에 영향을 줄 수 있습니다. RCP를 지원하는 AWS 서비스 목록을 포함하여 Organizations 및 RCP에 대한 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련될 때 AWS가 요청을 허용할지 여부를 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

AWS IoT Device Defender에서 IAM을 사용하는 방식

IAM을 사용하여 AWS IoT Device Defender에 대한 액세스를 관리하기 전에 AWS IoT Device Defender와 함께 사용할 수 있는 IAM 기능을 알아보세요.

AWS IoT Device Defender을 통해 사용할 수 있는 IAM 기능

IAM 특성	AWS IoT Device Defender 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키	예
ACLs	아니요
ABAC(정책 내 태그)	부분
임시 보안 인증	예
보안 주체 권한	예
서비스 역할	예
서비스 연결 역할	아니요

AWS IoT Device Defender 및 기타 AWS 서비스에서 대부분의 IAM 기능을 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서의 IAM으로 작동하는 AWS 서비스를 참조하세요.

AWS IoT Device Defender에 대한 ID 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 만드는 방법을 알아보려면 IAM 사용 설명서의 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 보안 인증 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

AWS IoT Device Defender 자격 증명 기반 정책 예시

AWS IoT Device Defender 자격 증명 기반 정책 예제를 보려면 AWS IoT Device Defender에 대한 자격 증명 기반 정책 예시 단원을 참조하세요.

AWS IoT Device Defender 내 리소스 기반 정책

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예시는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 또는 AWS 서비스가 포함될 수 있습니다.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 보안 주체와 리소스가 서로 다른 AWS 계정에 있는 경우, 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 엔터티(사용자 또는 역할)에도 리소스 액세스 권한을 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 크로스 계정 리소스 액세스를 참조하세요.

AWS IoT Device Defender 정책 작업

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 일반적으로 정책 작업의 이름은 연결된 AWSAPI 작업의 이름과 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

AWS IoT Device Defender 작업 목록을 보려면 서비스 권한 부여 참조를 참조하세요.

AWS IoT Device Defender의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      ":action1",
      ":action2"
         ]
```

AWS IoT Device Defender 자격 증명 기반 정책 예제를 보려면 AWS IoT Device Defender에 대한 자격 증명 기반 정책 예시 단원을 참조하세요.

AWS IoT Device Defender 정책 리소스

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 개체를 지정합니다. 문장에는 `Resource`또는 `NotResource`요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 태스크를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

AWS IoT Device Defender 리소스 유형 및 해당 ARN 목록을 보려면 서비스 권한 부여 참조를 참조하세요. 각 리소스의 ARN을 지정할 수 있는 태스크를 알아보려면 섹션을 참조하세요.

AWS IoT Device Defender 자격 증명 기반 정책 예제를 보려면 AWS IoT Device Defender에 대한 자격 증명 기반 정책 예시 단원을 참조하세요.

AWS IoT Device Defender 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우, AWS는 논리적 `AND` 태스크를 사용하여 평가합니다. 단일 조건 키의 여러 값을 지정하는 경우, AWS는 논리적 `OR` 태스크를 사용하여 조건을 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예컨대, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS는 전역 조건 키와 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

AWS IoT Device Defender 조건 키 목록을 보려면 서비스 권한 부여 참조를 참조하세요. 조건 키를 사용할 수 있는 태스크와 리소스를 알아보려면 섹션을 참조하세요.

AWS IoT Device Defender 자격 증명 기반 정책 예제를 보려면 AWS IoT Device Defender에 대한 자격 증명 기반 정책 예시 단원을 참조하세요.

AWS IoT Device Defender의 ACL

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

AWS IoT Device Defender를 사용한 ABAC

ABAC 지원(정책의 태그): 부분적

ABAC(속성 기반 액세스 제어)는 속성에 근거하여 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM 엔터티(사용자 또는 역할) 및 많은 AWS리소스에 태그를 연결할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

AWS IoT Device Defender에서 임시 보안 인증 정보 사용

임시 자격 증명 지원: 예

일부 AWS 서비스는 임시 보안 인증을 사용하여 로그인할 때 작동하지 않습니다. 임시 보안 인증으로 작동하는 AWS 서비스를 비롯한 추가 정보는 IAM 사용 설명서의 IAM으로 작업하는 AWS 서비스를 참조하십시오.

사용자 이름과 암호를 제외한 다른 방법을 사용하여 AWS Management Console에 로그인하면 임시 보안 인증을 사용하는 것입니다. 예컨대, 회사의 Single Sign-On(SSO) 링크를 사용하여 AWS에 액세스하면 해당 프로세스에서 자동으로 임시 자격 증명을 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 사용자에서 IAM 역할로 전환(콘솔)을 참조하세요.

AWS CLI 또는 AWS API를 사용하여 임시 보안 인증을 수동으로 만들 수 있습니다 그런 다음 이러한 임시 보안 인증을 사용하여 AWS에 액세스할 수 있습니다. AWS에서는 장기 액세스 키를 사용하는 대

신 임시 보안 인증을 동적으로 생성할 것을 권장합니다. 자세한 정보는 IAM의 임시 보안 자격 증명 단원을 참조하세요.

AWS IoT Device Defender의 서비스 간 보안 주체 권한

전달 액세스 세션(FAS) 지원: 예

IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 사람은 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 AWS 서비스를 직접 호출하는 보안 주체의 권한과 요청하는 AWS 서비스를 함께 사용하여 다운스트림 서비스에 대한 요청을 수행합니다. FAS 요청은 서비스에서 완료를 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 받은 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

AWS IoT Device Defender의 서비스 역할

서비스 역할 지원: 예

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 Create a role to delegate permissions to an AWS 서비스를 참조하세요.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 AWS IoT Device Defender 기능이 중단될 수 있습니다. AWS IoT Device Defender이 그 일을 하라는 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

AWS IoT Device Defender에 대한 서비스 연결 역할

서비스 링크 역할 지원: 아니요

서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 유형입니다. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 링크 역할은 AWS 계정에 나타나고, 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는 AWS 서비스를 참조하십시오. 서비스 연결 역할 열에서 Yes(이)가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 Yes(네) 링크를 선택합니다.

AWS IoT Device Defender에 대한 자격 증명 기반 정책 예시

기본적으로 사용자 및 역할에는 AWS IoT Device Defender 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface(AWS CLI) 또는 AWSAPI를 사용해 태스크를 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARN 형식을 포함하여 AWS IoT Device Defender에서 정의된 작업 및 리소스 유형에 대한 자세한 내용은 서비스 권한 부여 참조의 AWS IoT Device Defender에 대한 작업, 리소스 및 조건 키를 참조하세요.

주제

- 정책 모범 사례
- AWS IoT Device Defender 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 AWS IoT Device Defender 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책으로 시작하고 최소 권한을 향해 나아가기 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. AWS 계정에서 사용할 수 있습니다. 사용 사례에 고유한 AWS고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우 태스크를 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정

책 조건을 작성할 수 있습니다. AWS CloudFormation와 같이, 특정 AWS 서비스를 통해 사용되는 경우에만 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.

- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 – AWS 계정 계정에 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우, 추가 보안을 위해 MFA를 설정합니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

AWS IoT Device Defender 콘솔 사용

AWS IoT Device Defender 콘솔에 액세스하려면 최소한의 권한 세트가 있어야 합니다. 이러한 권한은 AWS 계정에서 AWS IoT Device Defender 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다. 최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 엔터티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 직접적으로 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요가 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 AWS IoT Device Defender 콘솔을 여전히 사용할 수 있도록 하려면 AWS IoT Device Defender *ConsoleAccess* 또는 *ReadOnly* AWS 관리형 정책을 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI나 AWSAPI를 사용하여 프로그래밍 방식으로 이 태스크를 완료할 수 있는 권한이 포함됩니다.

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

```
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS IoT Device Defender 보안 인증 및 액세스 문제 해결

다음 정보를 사용하여 AWS IoT Device Defender 및 IAM에서 발생할 수 있는 공통적인 문제를 진단하고 수정할 수 있습니다.

주제

- AWS IoT Device Defender에서 작업을 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 AWS 계정 외부의 사람이 내 AWS IoT Device Defender 리소스에 액세스할 수 있게 허용하기를 원합니다.

AWS IoT Device Defender에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 `mateojackson`IAM 사용자가 콘솔을 사용하여 가상 *`my-example-widget`*리소스에 대한 세부 정보를 보려고 하지만 가상 `:`*`GetWidget`*권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to
 perform: :GetWidget on resource: my-example-widget
```

이 경우 `:`*`GetWidget`*작업을 사용하여 *`my-example-widget`*리소스에 액세스할 수 있도록 `mateojackson`사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하십시오. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS IoT Device Defender에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 해당 서비스에 기존 역할을 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 AWS IoT Device Defender에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 `iam:PassRole` 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하십시오. 관리자는 로그인 보안 인증 정보를 제공하는 사람입니다.

내 AWS 계정 외부의 사람이 내 AWS IoT Device Defender 리소스에 액세스할 수 있게 허용하기를 원합니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제

어 목록(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- AWS IoT Device Defender에서 이러한 기능을 지원하는지 여부를 알아보려면 AWS IoT Device Defender에서 IAM을 사용하는 방식 단원을 참조하세요.
- 소유하고 있는 AWS 계정의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 자신이 소유한 다른 AWS 계정의 IAM 사용자에 대한 액세스 권한 제공을 참조하세요.
- 리소스에 대한 액세스 권한을 서드 파티 AWS 계정에게 제공하는 방법을 알아보려면 IAM 사용 설명서의 서드 파티가 소유한 AWS 계정에 대한 액세스 제공을 참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

AWS IoT Device Defender의 규정 준수 검증

AWS 서비스(이)가 특정 규정 준수 프로그램의 범위에 포함되는지 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반적인 정보는 AWS 규정 준수 프로그램을 참조하세요.

AWS Artifact(을)를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다.AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- 보안 규정 준수 및 거버넌스 - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.
- HIPAA 적격 서비스 참조 - HIPAA 적격 서비스가 나열되어 있습니다. 모든 AWS 서비스에 HIPAA 자격이 있는 것은 아닙니다.
- AWS 규정 준수 리소스 - 고객 조직이 속한 산업 및 위치에 적용될 수 있는 워크북 및 가이드 컬렉션입니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드에서는 AWS 서비스를 보호하기 위한 모범 사례를 요약하고 여러 프레임워크(미국 표준 기술 연구소(NIST),

결제 카드 산업 보안 표준 위원회(PCI), 국제 표준화기구(ISO) 등)에서 보안 컨트롤에 대한 지침을 매핑합니다.

- AWS Config 개발자 가이드의 규칙을 사용하여 리소스 평가 – AWS Config 서비스는 내부 사례, 산업 지침 및 규제에 대한 리소스 구성의 준수 상태를 평가합니다.
- AWS Security Hub – 이 AWS 서비스(은)는 AWS 내 보안 상태에 대한 포괄적인 보기를 제공합니다. Security Hub는 보안 컨트롤을 사용하여 AWS리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty - 이 AWS 서비스는 의심스럽고 악의적인 활동이 있는지 환경을 모니터링하여 AWS 계정, 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 탐지합니다. GuardDuty는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 따르는 데 도움을 줄 수 있습니다.
- AWS Audit Manager - 이 AWS 서비스은(는) AWS 사용을 지속해서 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 간소화할 수 있도록 지원합니다.

AWS IoT Device Defender의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전에서는 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 대기 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라 섹션을 참조하세요.

AWS 글로벌 인프라 외에도 AWS IoT Device Defender는 데이터 복원성과 백업 요구 사항을 지원하는 다양한 기능을 제공합니다.

AWS IoT Device Defender 사용 설명서 기록

다음 표에서는 AWS IoT Device Defender에 대한 문서 릴리스를 설명합니다.

변경 사항	설명	날짜
정식 버전	AWS IoT Device Defender의 최초 공개 릴리스입니다.	2023년 8월 2일
AWS IoT Device Defender가 이제 디바이스 연결 해제 기간 모니터링 지원	AWS IoT Device Defender Rules Detect가 이제 각 디바이스의 연결 해제 기간을 모니터링하는 새로운 연결 해제 기간 지표를 지원합니다. 이 추가 지표를 사용하면 디바이스 연결이 끊긴 시간을 추적하여 디바이스가 예상대로 작동하는지 확인할 수 있습니다. 또한 미리 정의된 임계값 수준에서 경보를 구성하고 디바이스 연결 문제가 지속되는 경우 알림을 받을 수 있습니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2023년 7월 20일
AWS IoT Device Defender 감사 기능이 IoT 정책의 잠재적 구성 오류 식별	감사 기능을 사용하여 결함을 식별하고, 문제를 해결하고, 필요한 수정 조치를 취하세요. 또한 이 새로운 기능은 디바이스가 의도하지 않은 리소스에 대한 액세스 권한을 얻을 수 있는 관대한 허용 문이 포함된 IoT 정책을 식별하는 데도 도움이 됩니다. 또한 와일드카드를 특정 문자열로 교체할 때 디바이스가 우회할 수 있는 거부 문에	2022년 12월 6일

	MQTT 와일드카드가 사용되는지도 검사합니다. 자세한 내용은 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	
AWS IoT Device Defender ML Detect 사용자 지정 지표 및 차원 지원	AWS IoT Device Defender이 이제 해지된 중간 인증 기관(CA)에 대한 새로운 감사 검사를 지원합니다. 중간 CA가 잠재적으로 손상되어 CA에 의해 해지되는 경우 해당 중간 CA에서 발급한 모든 인증서도 손상되어 유효하지 않을 수 있습니다. 이 새로운 감사 검사는 해지된 중간 CA에서 발급한 활성 디바이스 인증서를 식별하고 고객이 이러한 활성 디바이스 인증서를 검토하고 교체하는 데 도움이 됩니다. 자세한 내용은 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2022년 11월 10일

AWS IoT Device Defender ML Detect 사용자 지정 지표 및 차원 지원	ML Detect가 이제 플릿에 고유한 운영 상태 파라미터를 평가할 수 있는 사용자 지정 지표 모니터링을 지원합니다. Rules Detect로 정적 경보를 수동으로 설정하는 것 외에도 이제 기계 학습을 사용하여 사용자 지정 지표에 대한 플릿의 예상 동작을 자동으로 학습할 수 있습니다. 또한 ML Detect에 대한 새로운 차원 필터 지원을 통해 속성을 정의하여 ML 보안 프로필에서 더 정확한 지표를 평가할 수 있습니다. AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표	2022년 9월 14일
AWS IoT Device Management 및 AWS IoT Device Defender가 이제 ListMetricValues API를 통한 디바이스 지표 모니터링 지원	ListMetricValues API를 사용하여 보안 프로필에 속한 연결된 디바이스에서 과거 디바이스 측, 클라우드 측 및 사용자 지정 지표에 액세스할 수 있습니다. 이제 AWS IoT 관리 콘솔에서 데이터를 보는 것 외에도 프로그래밍 방식으로 모니터링하고 자체 시각화를 구축할 수 있는 유연성을 갖게 되었습니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2022년 4월 5일

AWS IoT Device Defender가 이제 경보 검증 상태 감지 지원	감지된 동작 이상에 대한 조사를 기반으로 경보를 검증합니다. 경보를 참양성, 양성, 가양성 또는 알 수 없음으로 검증하고, 검증의 설명을 제공할 수 있습니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2021년 9월 24일
AWS IoT Device Defender Audit One-Click 릴리스	Audit One-Click을 사용하면 AWS IoT Core 고객이 클릭 한 번으로 보안 모범 사례에 따라 계정 및 IoT 디바이스 감사를 시작하여 보안 기준을 쉽게 개선할 수 있습니다. Audit One-Click을 통해 고객은 사용 가능한 모든 감사 검사 및 일일 감사 일정을 활성화하는 등 사전 설정된 구성으로 AWS IoT Device Defender 감사를 설정할 수 있습니다. 또한 정기 보안 감사의 이점에 대한 상황에 맞는 설명도 제공합니다. Audit One-Click은 AWS IoT 콘솔에서만 사용할 수 있습니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2021년 9월 22일

AWS IoT Device Defender CloudFormation 지원	AWS IoT Device Defender Rules Detect는 이제 연결 해제 기간을 모니터링하는 새로운 연결 해제 기간 지표를 지원합니다. AWS IoT Device Defender는 이제 정기 감사 및 보안 프로필과 같은 AWS IoT Device Defender 리소스를 안전하고 효율적이며 반복 가능한 방식으로 생성하고 구성하는 데 AWS CloudFormation을 지원합니다. AWS IoT Device Defender가 지원하는 AWS CloudFormation 리소스 유형에 대해 자세히 알아보려면 IoT 리소스 유형 참조를 참조하세요.	2021년 3월 5일
AWS IoT Device Defender에 사용자 지정 지표에 대한 지원 추가	플릿 또는 사용 사례에 고유한 운영 상태 지표를 모니터링하는 데 AWS IoT Device Defender를 사용합니다. 경보는 Device Defender 콘솔에서 확인하거나 AWS Simple Notification Service(SNS)를 통해 공유할 수 있습니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2020년 12월 15일

AWS IoT Device Defender에 조사 결과 제외 시작 도입	조사 결과 제외 기능을 사용하면 보고 싶은 조사 결과를 선택하고 특정 리소스에 대해 규정을 준수하지 않는 조사 결과를 끌 수 있습니다. 또한 정의된 기간 동안 또는 무기한으로 조사 결과 제외를 구성할 수 있습니다. 설명서는 AWS IoT Device Defender개발자 안내서의 감사를 참조하세요.	2020년 8월 12일
AWS IoT Device Defender가 이제 주제 기반 지표 모니터링을 위한 차원 지원	차원 기능을 통해 고객은 Device Defender Detect가 평가하는 지표를 MQTT 주제별로 필터링할 수 있습니다. 차원은 수신된 메시지 수, 메시지 바이트 크기, 전송된 메시지 수, 소스 IP, 인증 실패 수와 같은 클라우드 측 지표를 지원합니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2020년 4월 2일
AWS IoT Device Defender ML Detect 정식 출시	AWS IoT Device Defender의 ML Detect 기능은 과거 데이터를 학습하여 플릿의 디바이스 수준 운영 및 보안 이상을 자동으로 감지합니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2020년 3월 24일

AWS IoT Device Defender 감사 기능에 네 가지 새로운 검사 추가	AWS IoT Device Defender 감사를 사용하여 사용 권한이 지나치게 관대하거나, 365일 이상 사용되지 않은 서비스에 액세스할 수 있거나, 예측 가능한 암호화 키가 있어 무차별 대입 공격에 취약한 것으로 확인된 Debian 기반 운영 체제에서 OpenSSL 버전을 사용하거나, RSA 키 생성을 잘못 처리하여 해킹에 취약한 것으로 확인된 Infineon RSA 라이브러리 버전을 사용하는 디바이스가 폴릿에 있는지 확인할 수 있습니다. 설명서는 AWS IoT Device Defender개발자 안내서의 감사를 참조하세요.	2019년 11월 25일
AWS IoT Device Defender가 감사 결과에 대한 완화 조치 지원	AWS IoT Device Defender에서 고객이 감사 조사 결과에 완화 조치를 적용할 수 있는 기능이 지원됩니다. 설명서는 AWS IoT Device Defender개발자 안내서의 감사를 참조하세요.	2019년 8월 6일
AWS IoT Device Defender가 등록되지 않은 디바이스의 동작 모니터링 지원	AWS IoT Core 레지스트리에 등록되지 않은 디바이스의 비정상적인 동작을 식별합니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2019년 5월 15일

AWS IoT Device Defender가 이제 통계적 이상 탐지 및 데이터 시각화 제공	통계적 이상 탐지를 사용하여 디바이스가 백분위수 기반 임계값 내에 있지 않을 경우 경고를 받을 수 있습니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2019년 2월 19일
AWS IoT Device Defender가 이제 디바이스 연결 해제 기간 모니터링 지원	AWS IoT Device Defender가 이제 두 개의 추가 클라우드 측 지표, 연결 시도 횟수, 연결 끊김 횟수를 지원합니다. 설명서는 AWS IoT Device Defender 개발자 안내서의 클라우드 측 지표를 참조하세요.	2018년 12월 19일