AWS IoT Device Defender 开发人员指南

AWS IoT Device Defender

AWS IoT Device Defender: AWS IoT Device Defender 开发人员指南

Table of Contents

什么是 AWS IoT Device Defender？ 1
您是 AWS IoT Device Defender 新用户吗？ 2
AWS IoT Device Defender 的工作原理 2
AWS IoT Device Defender 的特征 3
如何开始使用 AWS IoT Device Defender 4
相关服务 4
访问 AWS IoT Device Defender 4
AWS IoT Device Defender 定价 5
AWS IoT Device Defender 入门 6
设置 6
注册 AWS 账户 6
创建具有管理访问权限的用户 7
审计指南 8
先决条件 8
启用审计检查 8
查看审计结果 9
创建审计缓解操作 9
将缓解操作应用于审计查找结果 10
创建 AWS IoT Device Defender 审计 IAM 角色（可选） 10
启用 SNS 通知（可选） 11
启用日志记录（可选） 12
ML Detect 指南 12
先决条件 12
如何在控制台中使用 ML Detect 12
如何将 （ML Detect 与 CLI 一起使用 29
自定义查看 AWS IoT Device Defender 审计结果的时间和方式 43
开始使用 43
在控制台中自定义审计查找结果 44
在 CLI 中自定义您的审计查找结果 47
审核 54
问题严重性 54
后续步骤 55
审核检查 55
已吊销中间 CA 以进行活动设备证书检查 56

已吊销的 CA 证书仍处于活动状态 57
共享设备证书 58
设备证书密钥质量 59
CA 证书密钥质量 60
未经身份验证的 Cognito 角色过于宽容 62
经过身份验证的 Cognito 角色过于宽容 69
AWS IoT 策略过于宽容 78
AWS IoT 策略可能配置错误 83
角色别名过于宽容 87
角色别名允许访问未使用的服务 88
CA 证书即将过期 89
冲突的 MQTT 客户端 ID 90
设备证书即将过期 91
设备证书期限检查 93
已撤销的设备证书仍处于激活状态 94
日志记录已禁用 95
审核命令 95
管理审核设置 95
计划审核 102
运行按需审核 115
管理审核实例 116
检查审核结果 125
审计查找结果隐藏 134
审计查找结果隐藏的工作原理 134
如何在控制台中使用审计查找结果隐藏 134
如何在 CLI 中使用审计查找结果隐藏 142
审计查找结果隐藏 API 144
Detect 145
监控未注册设备的行为 146
安全使用案例 146
云端使用案例 147
设备端使用案例 149
概念 152
行为 153
ML Detect 156
ML Detect 的使用案例 156

ML Detect 的工作原理 157
最低要求 157
限制 158
在告警中标记误报和其它验证状态 158
受支持的指标 158
服务限额 159
ML Detect CLI 命令 159
ML Detect API 160
暂停或删除 ML Detect 安全配置文件 160
自定义指标 162
如何在控制台中使用自定义指标 162
如何使用 CLI 中的自定义指标 164
自定义指标 CLI 命令 168
自定义指标 API 169
设备端指标 169
输出字节数 (aws:all-bytes-out) 169
字节数 (aws:all-bytes-in) 171
侦听 TCP 端口计数 (aws:num-listening-tcp-ports) 172
侦听 UDP 端口计数 (aws:num-listening-udp-ports) 174
输出数据包数 (aws:all-packets-out) 175
数据包数 (aws:all-packets-in) 177
目标 IP (aws:destination-ip-addresses) 178
侦听 TCP 端口 (aws:listening-tcp-ports) 179
侦听 UDP 端口 (aws:listening-udp-ports) 180
已建立的 TCP 连接计数 (aws:num-established-tcp-connections) 180
设备指标文档规范 182
从设备发送指标 190
云端指标 191
消息大小 (aws:message-byte-size) 191
已发送的消息 (aws:num-messages-sent) 193
已收到的消息 (aws:num-messages-received) 194
授权失败 (aws:num-authorization-failures) 196
源 IP (aws:source-ip-address) 197
连接尝试 (aws:num-connection-attempts) 198
断开连接 (aws:num-disconnects) 199
断开连接持续时间（aws:disconnect-duration） 201

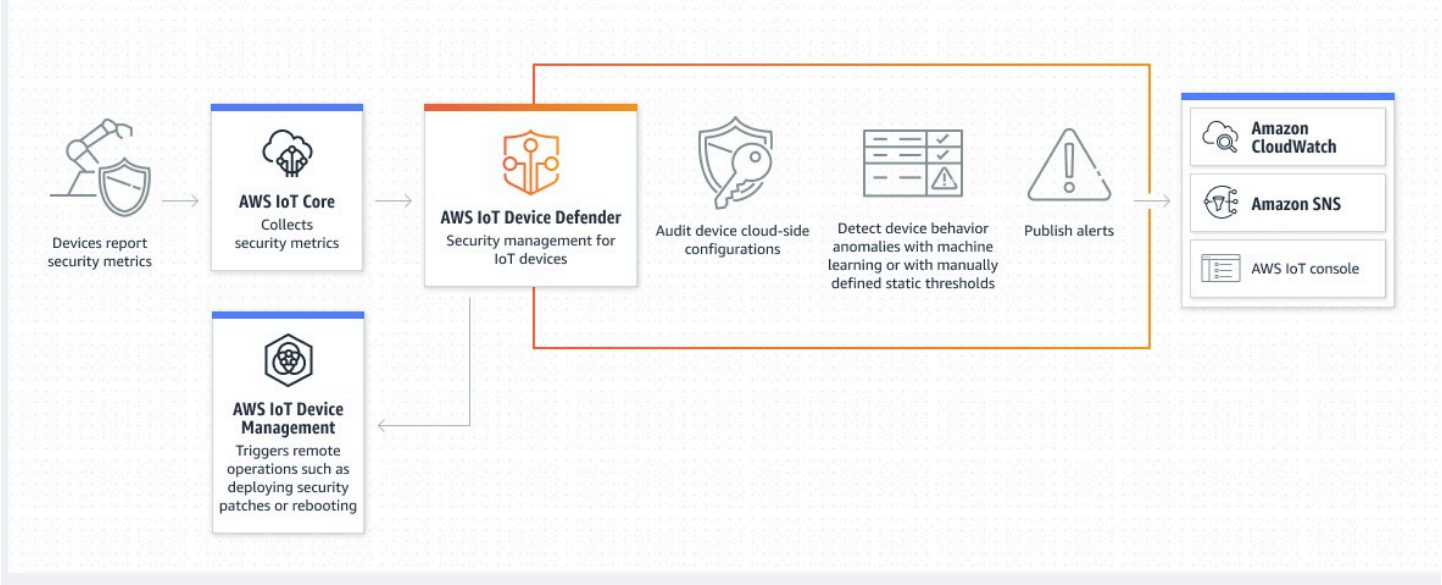
Detect 指标导出 201
检测指标导出的工作原理 204
指标导出模式 204
Detect 指标导出定价 205
权限 206
在 AWS IoT 控制台中设置 Detect 指标导出 207
创建安全配置文件来启用指标导出 209
更新安全配置文件来启用指标导出（CLI） 210
更新安全配置文件来关闭指标导出（CLI） 211
指标导出 CLI 命令 212
指标导出 API 操作 213
使用维度确定安全配置文件中指标的作用域 213
如何在控制台中使用维度 213
如何在 AWS CLI CLI 上使用维度 214
权限 219
授予 AWS IoT Device Defender Detect 向 SNS 主题发布告警的权限 219
Detect 命令 221
如何使用 AWS IoT Device Defender detect 223
缓解操作 225
审计缓解操作 225
检测缓解操作 229
如何定义和管理缓解操作 229
创建缓解操作 229
应用缓解操作 231
权限 236
缓解操作命令 242
将 AWS IoT Device Defender 与其它 AWS 产品结合使用 243
在运行 AWS IoT Greengrass 的设备上使用 AWS IoT Device Defender 243
将 AWS IoT Device Defender 与 FreeRTOS 和嵌入式设备搭配使用 243
配合使用 AWS IoT Device Defender和 AWS IoT Device Management 243
Security Hub 集成 244
启用和配置集成 244
AWS IoT Device Defender 将结果发送到 Security Hub 的方式 245
来自 AWS IoT Device Defender 的典型结果 247
停止 AWS IoT Device Defender 向 Security Hub 发送调查结果 252
防止跨服务混淆座席 252

设备代理的安全最佳实践 253
AWS IoT Device Defender 故障排除指南 256
安全性 261
数据保护 261
身份和访问管理 262
受众 262
使用身份进行身份验证 263
使用策略管理访问 265
AWS IoT Device Defender 如何与 IAM 协同工作 267
基于身份的策略示例 273
问题排查 275
合规性验证 277
弹性 278
文档历史记录 279

什么是 AWS IoT Device Defender？

使用 AWS IoT Device Defender 这项安全和监控服务来审计设备的配置，监控连接的设备，并降低安全风险。使用 AWS IoT Device Defender，您可以在整个 AWS IoT 设备队列中实施一致的安全策略，并能够在设备遭入侵时快速响应。IoT 实例集可能由大量具有不同功能、长期存在且地理位置分散的设备组成。这些特性导致实例集设置复杂且容易出错。由于设备的计算能力、内存和存储功能通常有限，因而限制了在设备本身上对加密和其他形式的安全功能的使用。

设备经常使用具有已知漏洞的软件。这些因素不仅使 IoT 队列成为吸引黑客的目标，而且导致难以持续保护设备队列安全。AWS IoT Device Defender 通过提供可识别安全问题以及与最佳实践的偏差的工具，解决了这些难题。AWS IoT Device Defender 可以审计设备队列，以确保它们遵守安全最佳实践并检测设备上的异常行为。下图显示了 AWS IoT Device Defender 的基本架构，以及它与 AWS IoT Core、Amazon CloudWatch 和 Amazon SNS 等服务如何相关。



主题

- 您是 AWS IoT Device Defender 新用户吗？
- AWS IoT Device Defender 的工作原理
- AWS IoT Device Defender 的特征
- 如何开始使用 AWS IoT Device Defender
- 相关服务
- 访问 AWS IoT Device Defender
- AWS IoT Device Defender 定价

您是 AWS IoT Device Defender 新用户吗？

如果您是首次接触 AWS IoT Device Defender 的用户，我们建议您先阅读以下部分：

- AWS IoT Device Defender 的工作原理
- AWS IoT Device Defender 的特征
- 如何开始使用 AWS IoT Device Defender
- 相关服务
- 访问 AWS IoT Device Defender
- AWS IoT Device Defender 定价

AWS IoT Device Defender 的工作原理

AWS IoT Device Defender 是一项完全托管式安全和监控服务，有助于您保护 IoT 设备队列。AWS IoT Device Defender 审计与您的设备关联的 IoT 资源，来确认它们符合安全最佳实践。如果检测到任何安全风险，审计检查功能会发出警报，并提供相关信息来协助缓解任何问题。AWS IoT Device Defender 还会持续监控来自云端和设备端的安全指标，来检测意外的设备行为，从而识别任何可能遭到入侵的设备。您可以按需或按计划启动审计检查，来评测您的 IoT 设备配置。

AWS IoT Device Defender 与 AWS IoT Core 结合使用来整合设备交互的上下文，从而提高审计检查的准确性。AWS IoT Device Defender 收集和分析来自所连接设备的高价值安全指标，来检测异常行为。使用 Rules Detect 时，将根据用户定义的行为持续评估指标数据。使用 ML Detect 时，自动构建的机器学习（ML）模型会持续评估指标数据来识别异常。

计划审计任务的结果和任何检测到的设备活动异常都会发布到 AWS IoT 控制台和 AWS IoT Device Defender API。可通过 Amazon CloudWatch 访问这些内容。此外，您可以将 AWS IoT Device Defender 配置为将结果发送到 Amazon SNS 主题，以便与安全控制面板集成或启动自动修复工作流程。

AWS IoT Device Defender 支持各种使用案例，包括：

- 保护您的设备：您可以根据 AWS IoT 安全最佳实践审计与设备相关的资源，来协助您检测设备漏洞。AWS IoT Device Defender 审计有助于您识别和发现设备面临的风险，并确认安全措施已到位。
- 检测异常设备行为：您可以精确确定连接模式的变化，揭示设备与未经授权的端点的通信，并识别入站和出站设备流量模式的变化。
- 获取见解来降低风险：您可以采取措施来缓解审计查找结果或 Detect 警报中发现的问题。

- 维护和维持设备安全：您可以使用审计和检测检查中的见解来诊断和修复可能的安全漏洞。
- 增强设备安全性：您可以区分配置不当的设备，探测设备队列的运行状况，并找到意外的设备行为指标。

AWS IoT Device Defender 的特征

以下是 AWS IoT Device Defender 的一些关键功能。

主要功能

审核	AWS IoT Device Defender 根据《IAM 用户指南》中的 AWS IoT 安全最佳实践审计您的设备相关资源。AWS IoT Device Defender 报告不符合安全最佳实践的配置，例如过度宽松的权限策略，此类策略可能允许一台设备读取和更新许多其它设备的数据。
Rules Detect	AWS IoT Device Defender 通过持续监控设备和 AWS IoT Core 中的高价值安全指标，检测可能表明遭入侵的异常设备行为。您可以通过为这些指标设置行为（规则），为一组设备指定正常的设备行为。AWS IoT Device Defender 根据用户定义的行为（规则）监控和评估为这些指标报告的每个数据点，并在检测到异常时向您发出警报。
ML Detect	AWS IoT Device Defender 使用机器学习（ML）模型自动为您设置设备行为，此类模型使用后续 14 天的六个云端指标和七个设备端指标的设备数据。然后，该服务每天对模型进行重新训练（只要它有足够的数据来训练模型），根据最初模型构建后的最新后续 14 天的数据刷新预期的设备行为。AWS IoT Device Defender 使用机器学习模型监控和识别这些指标的异常数据点，并在检测到异常时触发警报。

提示	AWS IoT Device Defender 向 AWS IoT 控制台、Amazon CloudWatch 和 Amazon SNS 发布警报。
缓解方法	可以使用 AWS IoT Device Defender 通过提供有关设备的上下文和历史信息（例如设备元数据、设备统计数据和设备的历史警报）来调查问题。还可以使用 AWS IoT Device Defender 内置的缓解操作对审计和检测警报执行缓解步骤，例如，将事物添加到事物组、替换默认策略版本和更新设备证书。

如何开始使用 AWS IoT Device Defender

有关开始使用 AWS IoT Device Defender 的帮助信息，请参阅以下教程。

- 设置
- ML Detect 指南
- 审计指南
- 自定义查看 AWS IoT Device Defender 审计结果的时间和方式

相关服务

- AWS IoT Greengrass：AWS IoT Greengrass 提供与 AWS IoT Device Defender 的预构建集成来持续监控设备行为。
- AWS IoT Device Management：您可以使用 AWS IoT Device Management 队列索引来搜索、聚合 AWS IoT Device Defender 检测违规情况并编制索引。

访问 AWS IoT Device Defender

您可以使用 AWS IoT Device Defender 控制台或 API 访问 AWS IoT Device Defender。

AWS IoT Device Defender 定价

对于 AWS IoT Device Defender，您只需按实际使用量付费。没有最低费用，也不会强制使用服务。但是，审计和检测功能需要单独付费。审计定价是按每月、每台设备确定的。开启审计功能后，系统会根据一个月内处于活动状态的设备主体数量向您收费。因此，在使用此特征时，添加或删除审计检查不会影响您的月度账单。您可以使用 AWS 定价计算器在单一估算中计算您的 AWS IoT Device Defender 和架构成本。

- AWS 定价计算器

开始使用 AWS IoT Device Defender

您可以使用以下教程来学习使用 AWS IoT Device Defender。

主题

- 设置
- 审计指南
- ML Detect 指南
- 自定义查看 AWS IoT Device Defender 审计结果的时间和方式

设置

首次使用 AWS IoT Device Defender 前，请完成以下任务：

主题

- 注册 AWS 账户
- 创建具有管理访问权限的用户

注册 AWS 账户

如果您还没有 AWS 账户，请完成以下步骤来创建一个。

注册 AWS 账户

1. 打开 https://portal.aws.amazon.com/billing/signup。
2. 按照屏幕上的说明进行操作。

   在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

   当您注册 AWS 账户时，系统将会创建一个 AWS 账户根用户。根用户有权访问该账户中的所有 AWS 服务 和资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行需要根用户访问权限的任务。

注册过程完成后，AWS 会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 https://aws.amazon.com/ 并选择我的账户来查看当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

注册 AWS 账户 后，请保护好您的 AWS 账户根用户，启用 AWS IAM Identity Center，并创建一个管理用户，以避免使用根用户执行日常任务。

保护您的 AWS 账户根用户

1. 选择根用户并输入您的 AWS 账户电子邮件地址，以账户拥有者身份登录 AWS Management Console。在下一页上，输入您的密码。

   要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的以根用户身份登录。
2. 为您的根用户启用多重身份验证 (MFA)。

   有关说明，请参阅《IAM 用户指南》中的为 AWS 账户 根用户启用虚拟 MFA 设备（控制台）。

创建具有管理访问权限的用户

1. 启用 IAM Identity Center。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的启用 AWS IAM Identity Center。
2. 在 IAM Identity Center 中，为用户授予管理访问权限。

   有关如何使用 IAM Identity Center 目录 作为身份源的教程，请参阅《AWS IAM Identity Center 用户指南》中的使用默认的 IAM Identity Center 目录 配置用户访问权限。

以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

  要获取使用 IAM Identity Center 用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的登录 AWS 访问门户。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的创建权限集。
2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的添加组。

这些任务创建一个 AWS 账户和一个具有该账户的管理员权限的用户。

审计指南

本教程提供有关如何配置定期审计、设置告警、查看审计结果和减少审计问题的说明。

主题

- 先决条件
- 启用审计检查
- 查看审计结果
- 创建审计缓解操作
- 将缓解操作应用于审计查找结果
- 创建 AWS IoT Device Defender 审计 IAM 角色（可选）
- 启用 SNS 通知（可选）
- 启用日志记录（可选）

先决条件

要完成本教程，您需要：

- AWS 账户。如果您尚未拥有账户，请参阅设置。

启用审计检查

在以下流程中，您可以启用审计检查以查看账户和设备设置及策略，以确保安全措施已就绪。在本教程中，我们指导您启用所有审计检查，但您可以仅选择所需的检查。

审计定价是按每月每台设备计数制定的（连接到 AWS IoT 机群设备)。因此，在使用此特征时，添加或删除审计检查不会影响您的月度账单。

1. 打开AWS IoT控制台。在导航窗格中，展开安全并选择简介。
2. 选择自动执行 AWS IoT 安全审计。审计检查会自动开启。

3. 展开审计，然后选择设置以查看您的审计检查。选择审计检查名称以了解审计检查的作用。有关审计检查的详细信息，请参阅审核检查。
4. （可选）如果您已经有要使用的角色，请选择管理服务权限，从列表中选择该角色，然后选择更新。

查看审计结果

以下流程介绍如何查看审计结果。在本教程中，您将看到在 启用审计检查 教程设置的审计检查的审计结果。

查看审计结果

1. 打开AWS IoT控制台。在导航窗格中，依次展开安全、审计，然后选择结果。
2. 选择您想要调查的审计计划的名称。
3. 在不合规检查中，在缓解下，选择信息按钮以获取有关其不合规原因的信息。有关如何使您的不合规检查变为合规的指导，请参阅审核检查。

创建审计缓解操作

在以下过程中，您将创建一个 AWS IoT Device Defender 审计缓解操作以启用 AWS IoT 日志记录。每个审计检查都已映射缓解操作，这些操作将影响您为想要修复的审计检查所选择的操作类型。有关更多信息，请参阅缓解操作。

使用 AWS IoT 控制台创建缓解操作

1. 打开AWS IoT控制台。在导航窗格中，依次展开安全、检测，然后选择缓解操作。
2. 在 Mitigation Actions（缓解操作）页面上，选择 Create（创建）。
3. 在创建新的缓解操作页面上，对于操作名称，为您的缓解操作输入唯一名称，例如 *EnableErrorLoggingAction*。
4. 对于操作类型，选择启用 AWS IoT 日志记录。
5. 在权限中，选择创建角色。对于角色名称，使用 *IoTMitigationActionErrorLoggingRole*。然后选择 Create。
6. 在参数中，在用于日志记录的角色下，选择 `IoTMitigationActionErrorLoggingRole`。对于 Log level（日志级别），选择 `Error`。
7. 选择创建。

将缓解操作应用于审计查找结果

以下流程介绍如何将缓解操作应用于审计结果。

减少不合规审计查找结果

1. 打开AWS IoT控制台。在导航窗格中，依次展开安全、审计，然后选择结果。
2. 选择要响应的审计结果。
3. 检查您的结果。
4. 选择 Start mitigation actions（启动缓解操作）。
5. 对于日志记录已禁用，请选择您之前创建的缓解操作 `EnableErrorLoggingAction`。您可以为每个不合规的调查发现选择适当的操作以解决这些问题。
6. 在选择原因代码中，选择审计检查返回的原因代码。
7. 选择启动任务。缓解操作可能需要几分钟时间来运行。

要检查缓解操作是否有效

1. 在 AWS IoT 控制台的导航窗格中，选择设置。
2. 在服务日志中，确认日志级别为 `Error (least verbosity)`。

创建 AWS IoT Device Defender 审计 IAM 角色（可选）

在以下过程中，您创建一个 AWS IoT Device Defender 审计 IAM 角色，为 AWS IoT Device Defender 提供对于 AWS IoT 的读取访问权限。

创建用于 AWS IoT Device Defender 的服务角色（IAM 控制台）

1. 登录 AWS Management Console，然后通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在 IAM 控制台的导航窗格中，选择角色，然后选择创建角色。
3. 选择 AWS 服务 角色类型。
4. 在其它 AWS 服务的用例中，选择 AWS IoT，然后选择 IoT - Device Defender 审计。
5. 选择下一步。
6. （可选）设置权限边界。这是一项高级特征，可用于服务角色，但不可用于服务相关角色。

展开权限边界部分，然后选择使用权限边界控制最大角色权限。IAM 包括您的账户中的 AWS 托管式策略和客户管理型策略的列表。选择要用于权限边界的策略，或选择创建策略以打开新的浏览器选项卡并从头开始创建新策略。有关更多信息，请参阅《IAM 用户指南》中的创建 IAM policy。在您创建策略后，关闭该选项卡并返回到您的原始选项卡，以选择要用于权限边界的策略。

7. 选择下一步。
8. 输入有助于标识此角色的作用的角色名称。角色名称在您的 AWS 账户 内必须是唯一的。名称不区分大小写。例如，您无法同时创建名为 **PRODROLE** 和 **prodrole** 的角色。由于多个实体可能引用该角色，因此，角色创建完毕后，您将无法编辑角色名称。
9. （可选）对于描述，输入新角色的描述。
10. 在 Step 1: Select trusted entities（步骤 1：选择可信实体）或 Step 2: Select permissions（步骤 2：选择权限）部分中的 Edit（编辑），以编辑角色的用户案例和权限。
11. （可选）通过以键值对的形式附加标签来向用户添加元数据。有关在 IAM 中使用标签的更多信息，请参阅《IAM 用户指南》中的标记 IAM 资源。
12. 检查角色，然后选择 Create role。

启用 SNS 通知（可选）

在以下过程中，您可以启用 Amazon SNS（SNS）通知，以便在审计发现任何不合规的资源时向您发出警报。在本教程中，您将为 启用审计检查 教程中启用的审计检查设置通知。

1. 附加允许通过 AWS Management Console访问 SNS 的策略（如果您尚未附加）。为此，您可以按照《IAM 用户指南》的将策略附加到 IAM 用户组中的说明操作，并选择 AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction 策略。
2. 打开AWS IoT控制台。在导航窗格中，依次展开安全、审计，然后选择设置。
3. 在 Device Defender 审计设置页面的底部，选择启用 SNS 警报。
4. 选择 Enabled (已启用)。
5. 对于主题，选择创建新主题。将主题命名为 *IoTDDNotifications*，然后选择创建。对于角色，选择您在创建 AWS IoT Device Defender 审计 IAM 角色（可选）中创建的角色。
6. 选择更新。
7. 如果您希望通过 Amazon SNS 在运维平台上接收电子邮件或文本，请参阅使用 Amazon Simple Notification Service 发送用户通知。

启用日志记录（可选）

本流程介绍如何启用 AWS IoT 将信息记录到 CloudWatch Logs 中。这将允许您查看您的审计结果。启用日志记录可能会导致费用产生。

要启用日志记录：

1. 打开AWS IoT控制台。在导航窗格上，选择设置。
2. 在日志中，选择管理日志。
3. 对于选择角色，选择创建角色。将角色命名为 *AWSIoTLoggingRole*，然后选择创建。此时会自动附加策略。
4. 对于日志级别，选择调试(最详细)。
5. 选择更新。

ML Detect 指南

在本入门指南中，您将创建一个 ML Detect 安全配置文件，该配置文件使用机器学习 (ML) 根据设备中的历史指标数据创建预期行为模型。当 ML Detect 正在创建 ML 模型时，您可以监控其进度。构建 ML 模型后，您可以持续查看和调查告警，并缓解已发现的问题。

有关 ML Detect 及其 API 和 CLI 命令的更多信息，请参阅 ML Detect。

本章包含以下部分：

- 先决条件
- 如何在控制台中使用 ML Detect
- 如何将（ML Detect 与 CLI 一起使用

先决条件

- AWS 账户。如果您尚未拥有账户，请参阅设置。

如何在控制台中使用 ML Detect

教程

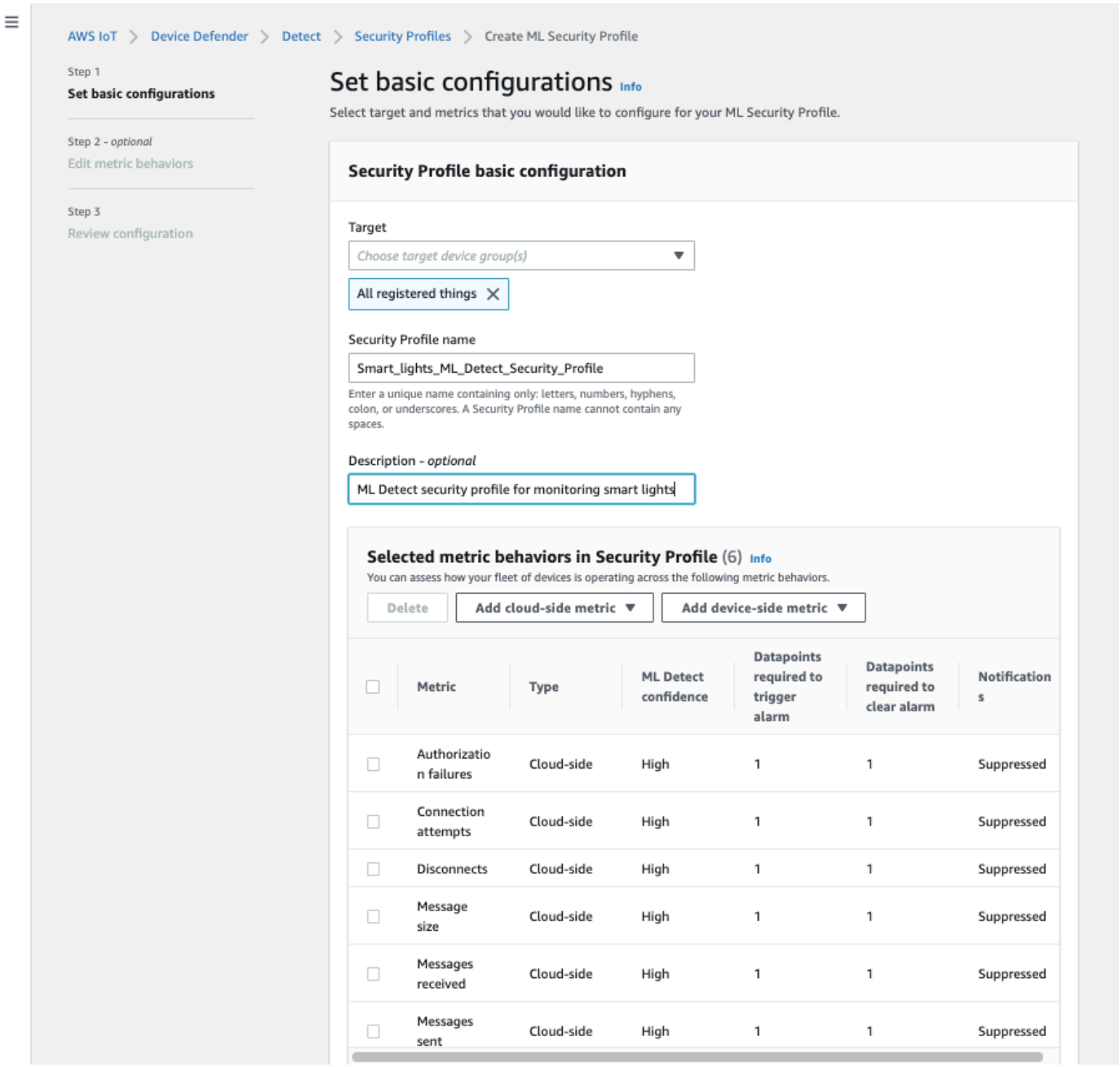
- 启用 ML Detect

- 监控您的 ML 模型状态
- 查看您的 ML Detect 告警
- 微调您的机器学习（ML）告警
- 标记告警的验证状态
- 缓解已确定的设备问题

启用 ML Detect

以下流程详细介绍了如何在控制台中设置 ML Detect。

1. 首先，确保您的设备将按 ML Detect 最低要求中的定义要求创建最小的数据点，以进行持续训练和刷新模型。要进行数据收集，请确保您的安全配置文件已附加到目标，该目标可以是事物或事物组。
2. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护）。选择 Detect（检测）、Security profiles（安全配置文件）、Create security profile（创建安全配置文件），然后选择 Create ML anomaly Detect profile（创建 ML 异常检测配置文件）。
3. 在 Set basic configurations（设置基本配置）页面上，执行以下操作。
   - 在 Target（目标）项下，选择您的目标设备组。
   - 在 Security profile name（安全配置文件名称）中，输入您的安全配置文件的名称。
   - （可选）在 Description（说明）项下，您可以编写 ML 配置文件的简述。
   - 在 Selected metric behaviors in Security Profile（安全配置文件中特定的指标行为）项下，选择要监控的指标。



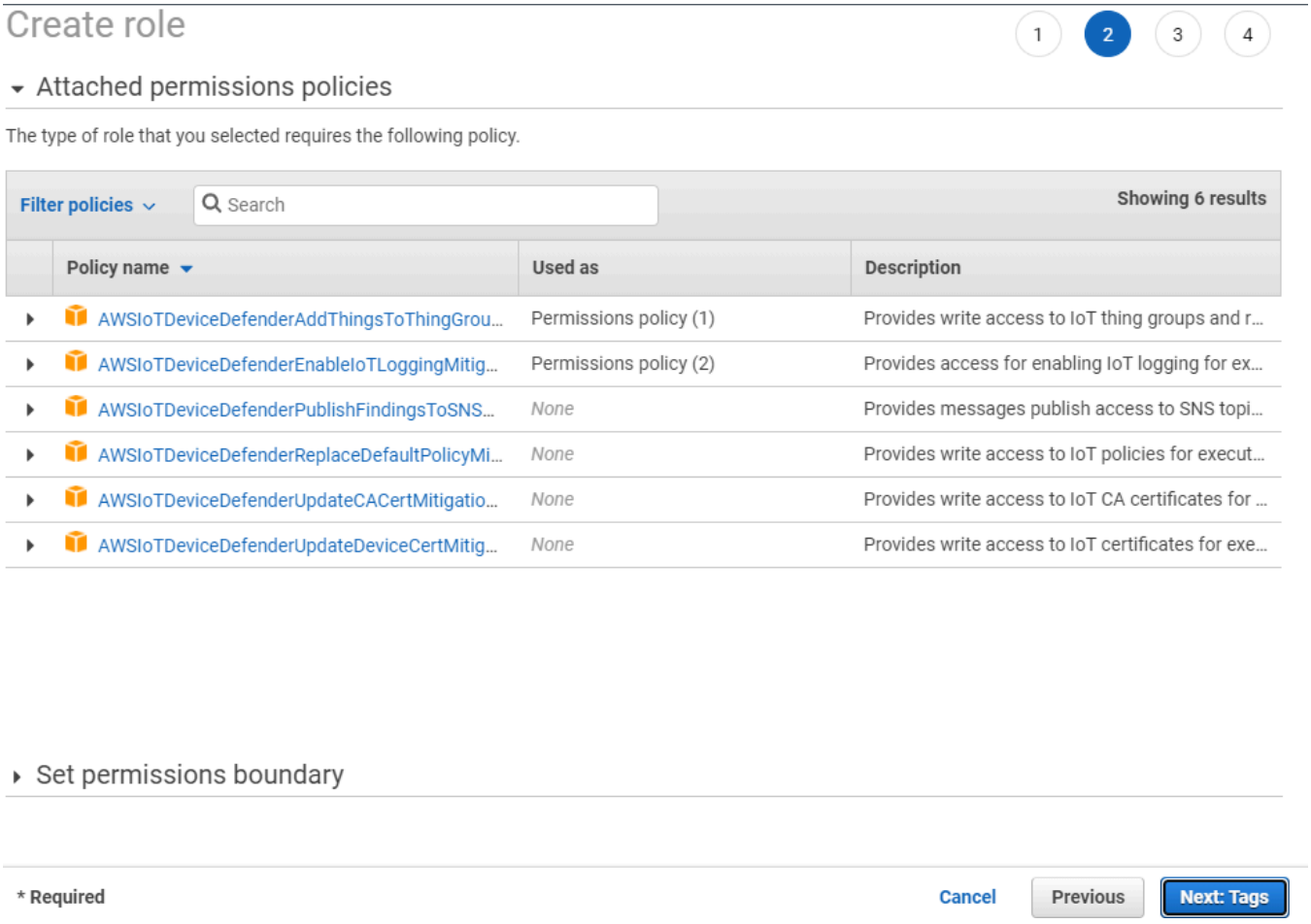
完成此操作后，选择 Next (下一步)。

4. 在 Set SNS (optional)（设置 SNS（可选））页面上，为设备违反配置文件中的行为时的告警通知指定 SNS 主题。选择要用于发布到选定 SNS 主题的 IAM 角色。

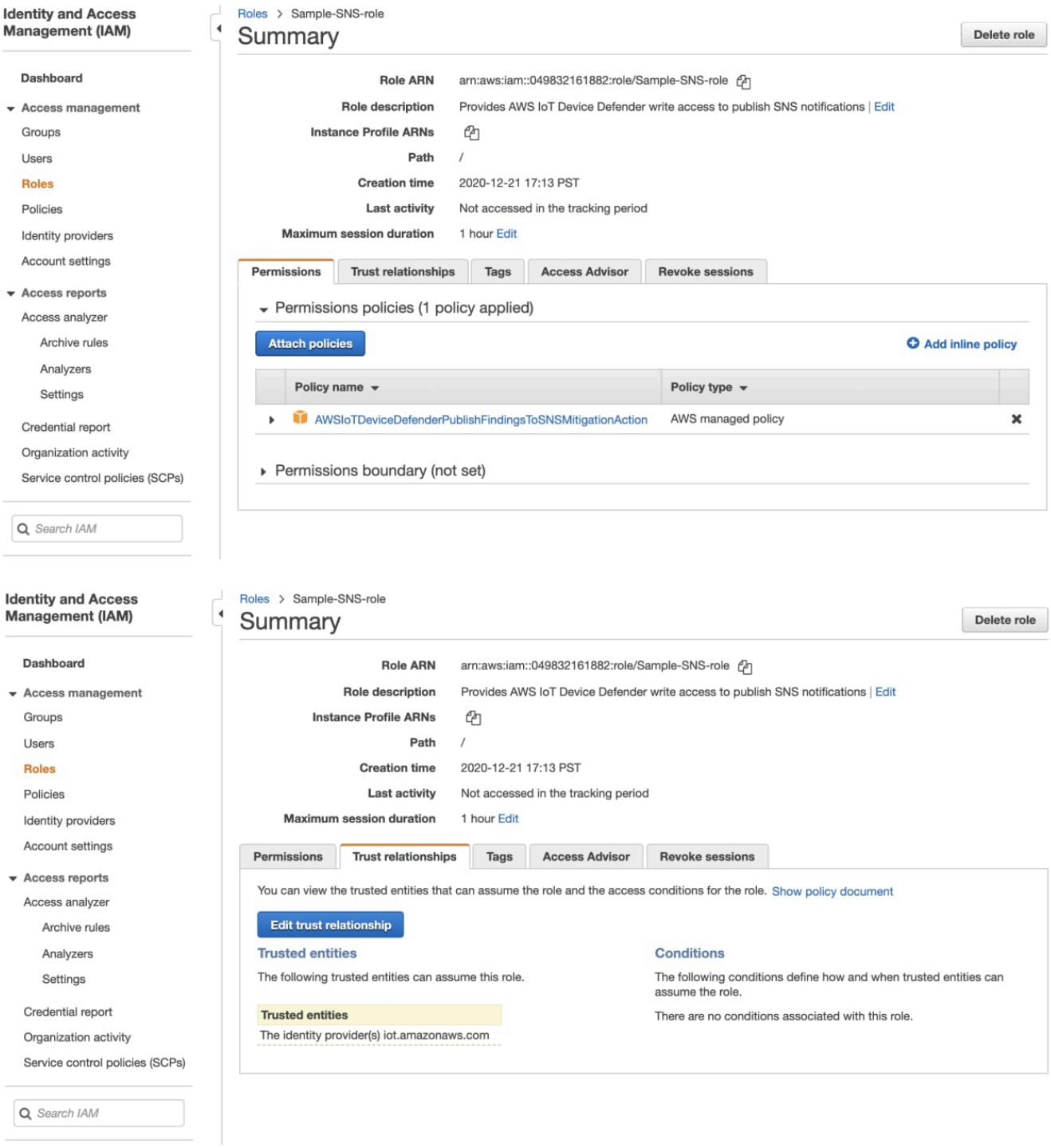
   如果您还没有 SNS 角色，请使用以下步骤创建具有所需的适当权限和信任关系的角色。

   - 导航到 IAM 控制台。在导航窗格中，选择角色，然后选择创建角色。

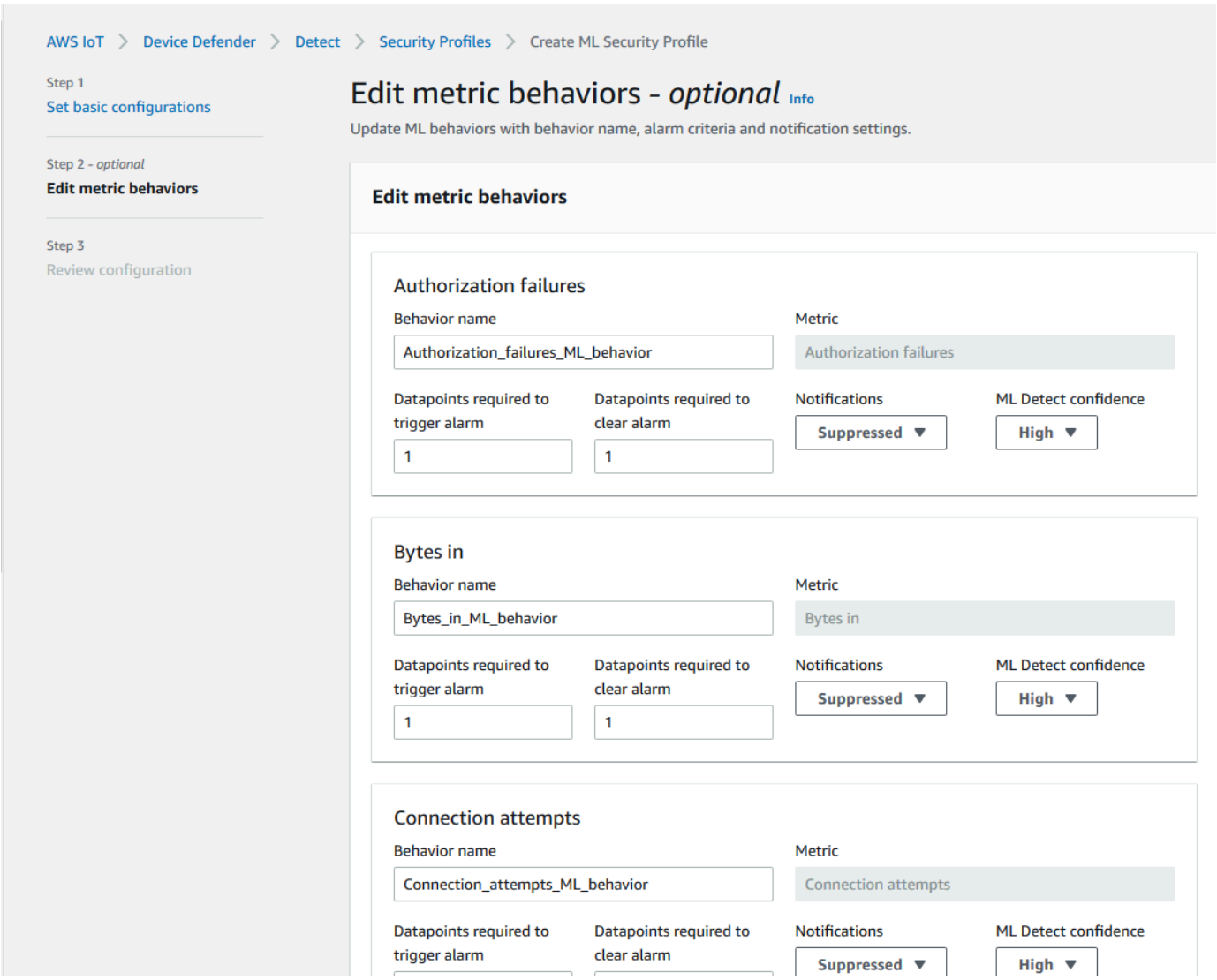
- 在Select type of trusted entity（选择信任实体的类型）项下，选择 AWS service（亚马逊云科技服务）。然后，在 Choose a use case（选择使用案例）中，选择 IoT，在 Select your use case（选择您的使用案例）项下，选择 IoT - Device Defender Mitigation Actions（IoT - Device Defender 缓解操作）。完成操作后，选择 Next: Permissions（下一步：权限）。
- 在 Attached permissions policies（附加权限策略）项下，确保选择了 AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction，然后选择 Next: Tags（下一步：标签）。



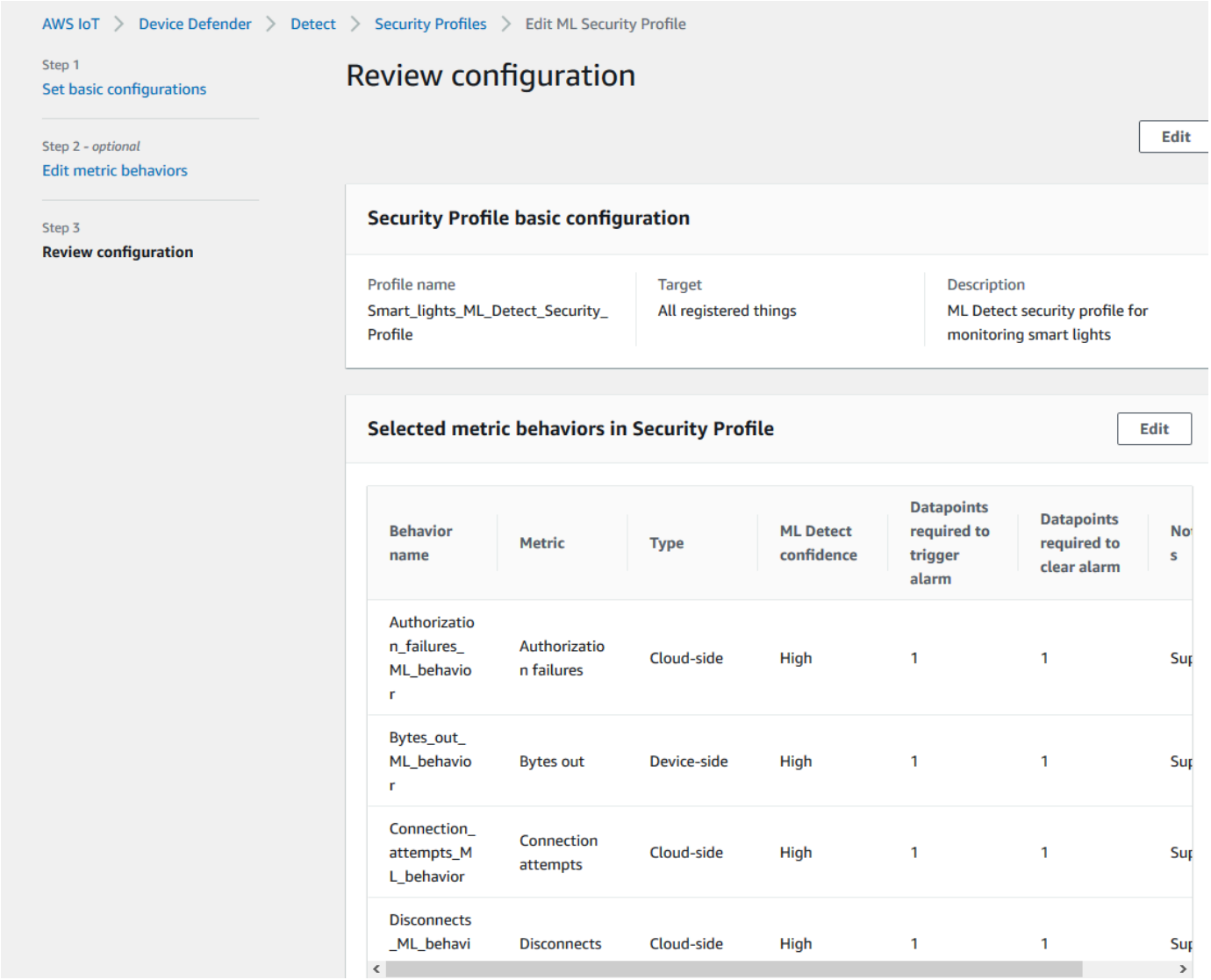
- 在 Add tags (optional)（添加标签（可选））项下，您可以添加要与角色关联的任何标签。完成此操作后，选择 Next: Review (下一步：审核)。
- 在 Review（审核）项下，请为您的角色指定一个名称，并确保 AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction 列示在 Permissions（权限）项下，且 AWS service: iot.amazonaws.com 列在 Trust relationships（信任关系）项下。完成后，选择 Create role（创建角色）。



5. 在 Edit Metric behavior（编辑指标行为）页面上，您可以自定义您的机器学习（ML）行为设置。



6. 完成此操作后，选择 Next (下一步)。
7. 在 Review configuration（查看配置）页面上，验证您希望机器学习监控的行为，然后选择 Next（下一步）。



8. 创建了安全配置文件之后，您将会被重定向到 Security Profiles（安全配置文件）页面，此处将显示新创建的安全配置文件。

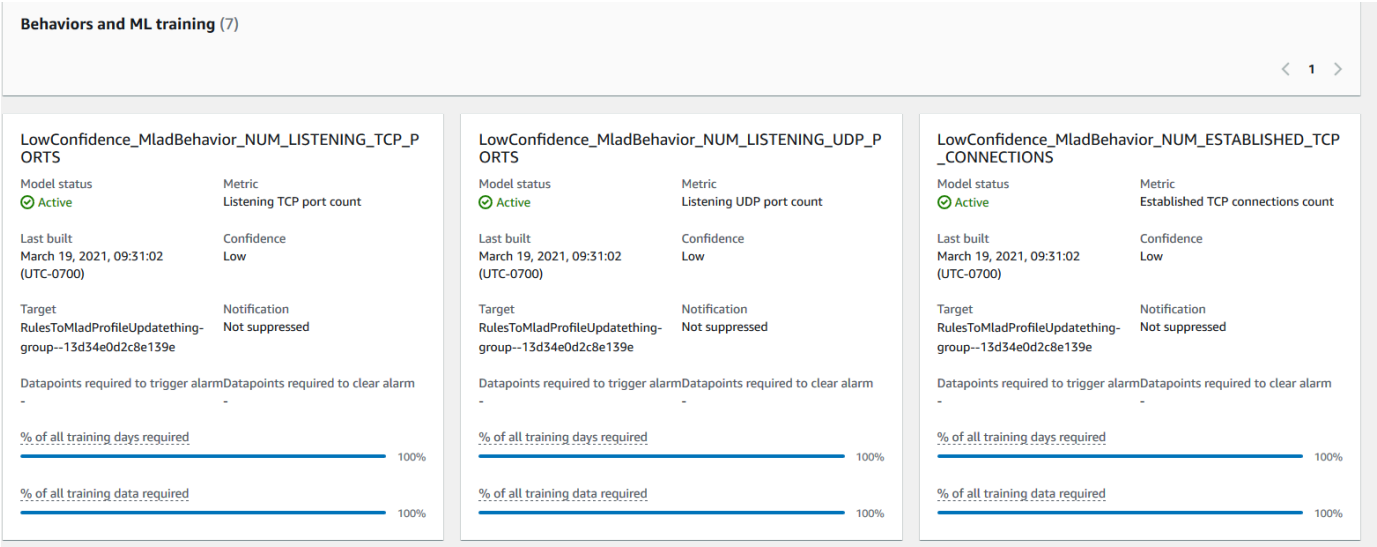
> **Note**
>
> 初始 ML 模型训练和创建需要 14 天才能完成。如果您的设备上存在任何异常活动，则在完成后您应会看到告警。

监控您的 ML 模型状态

当您的 ML 模型处于初始训练阶段时，您可以通过执行以下步骤随时监控其进度。

1. 在 AWS IoT 控制台中的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Security profiles（安全配置文件）。
2. 在 Security Profiles（安全配置文件）页面上，选择您要查看的安全配置文件。然后，选择 Behaviors and ML training（行为和机器学习（ML）训练）。
3. 在 Behaviors and ML training（行为和机器学习（ML）训练）页面上，检查 ML 模型的训练进度。

   在模型状态为 Active（激活）后，它将开始根据您的使用情况做出 Detect 决策，并每天更新配置文件。

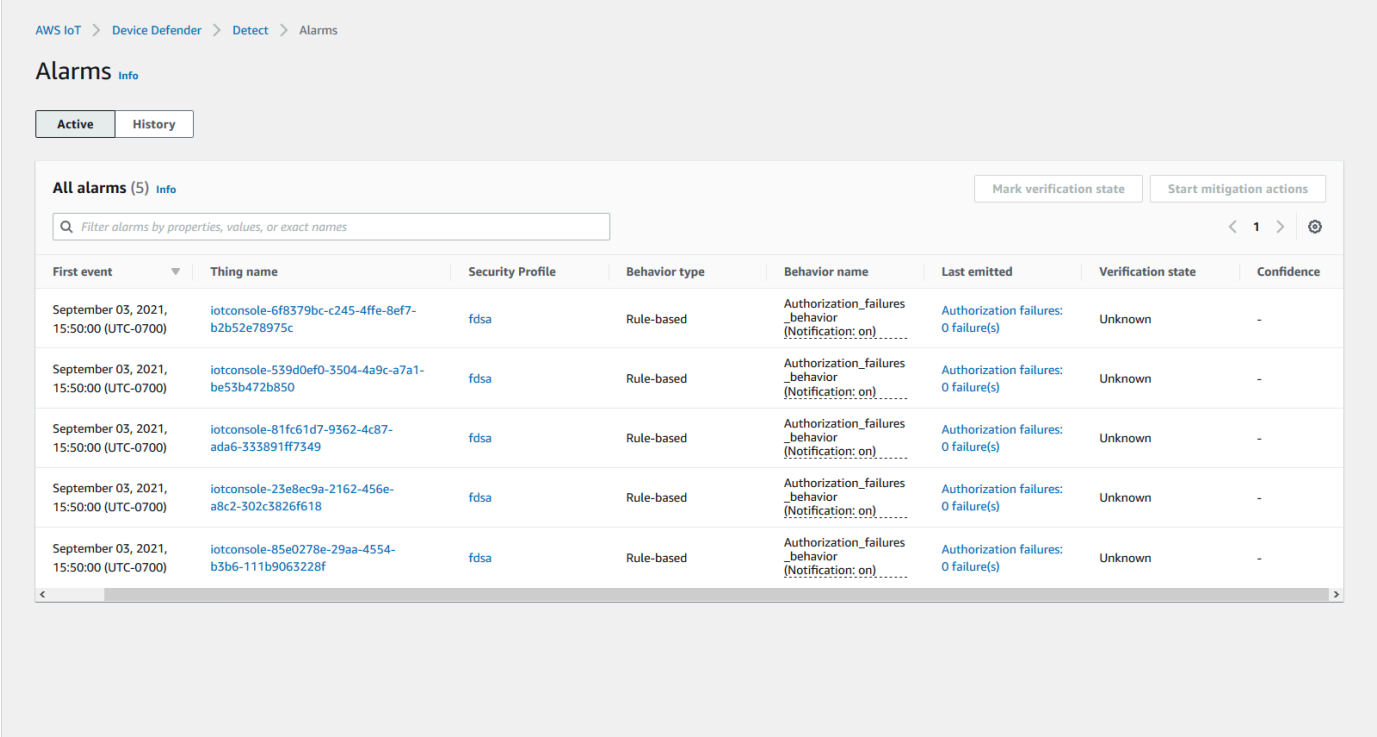


> **Note**
>
> 如果您的模型未按预期进行，请确保您的设备满足 最低要求 的要求。

查看您的 ML Detect 告警

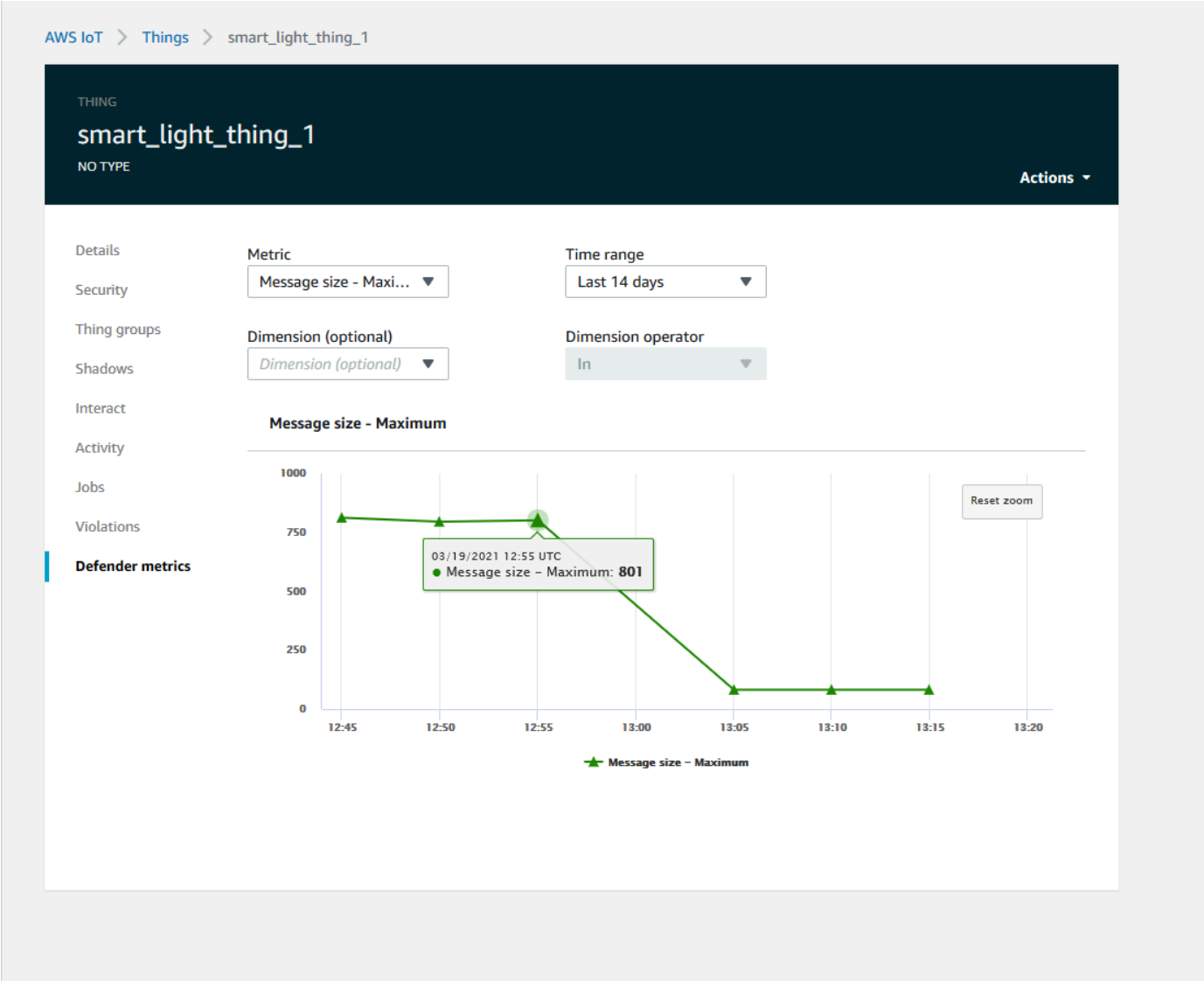
在构建 ML 模型并准备进行数据推理之后，您可以定期查看和调查由模型识别的告警。

1. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Alarms（告警）。



2. 如果您导航到 History（历史记录）选项卡上，您还可以查看不再处于告警状态的设备相关详细信息。

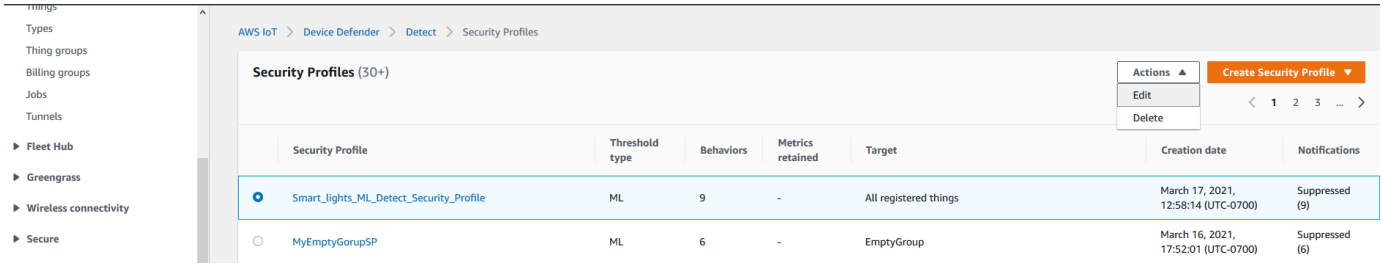
若要获取更多信息，请在 Manage（管理）项下选择 Things（事物），选择您想要查看更多详细信息的事物，然后导航到 Defender metrics（Defender 指标）。您可以访问 Defender metrics graph（Defender 指标图形）并对 Active（激活）选项卡中告警的任何内容进行调查。在这种情况下，图形显示了启动告警的消息大小峰值。您可以看到告警随后被清除。



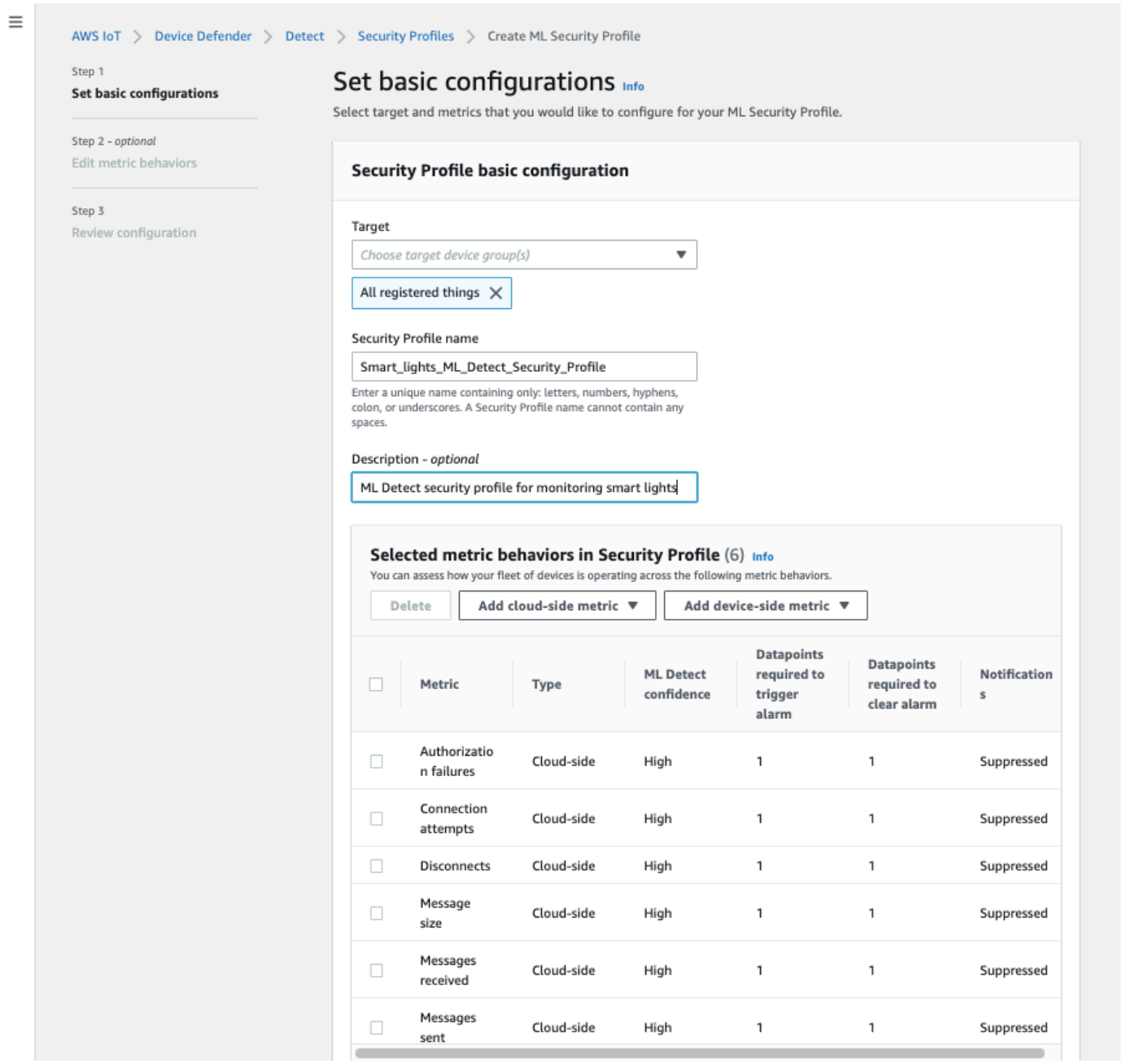
微调您的机器学习（ML）告警

在构建 ML 模型并准备好进行数据评估之后，您可以更新安全配置文件的机器学习（ML）行为设置以更改配置。以下流程介绍如何在 AWS CLI 中更新您的安全配置文件的机器学习（ML）行为设置。

1. 在 AWS IoT 控制台中的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Security profiles（安全配置文件）。
2. 在 Security Profiles（安全配置文件）页面上，选中要查看的安全配置文件旁边的复选框。然后依次选择 Actions（操作）、Edit（编辑）。

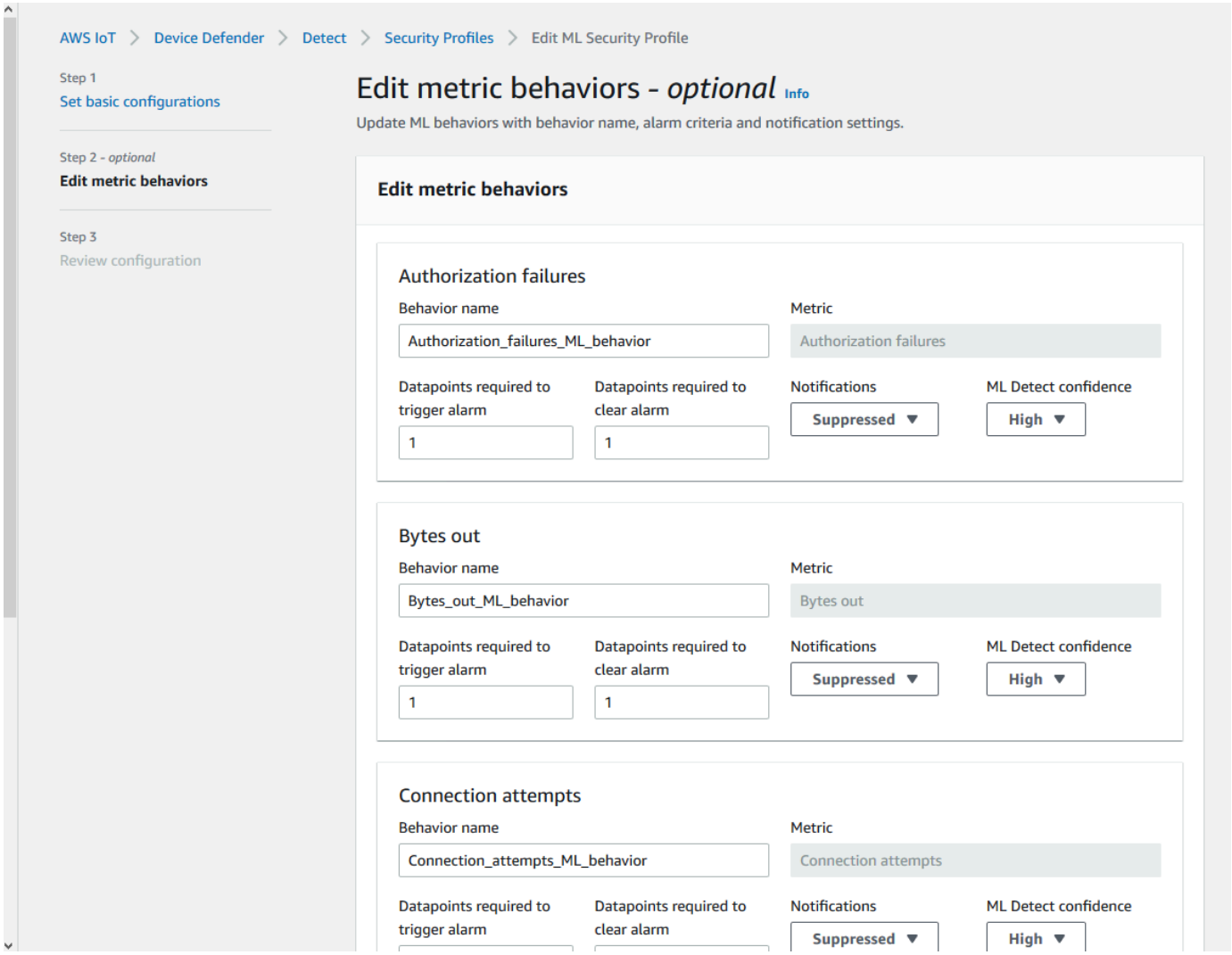


3. 在 Set basic configurations（设置基本配置）项下，您可以调整安全配置文件目标事物组或更改要监控的指标。



4. 您可以通过导航到 Edit metric behaviors（编辑指标行为）更新以下任意一项。

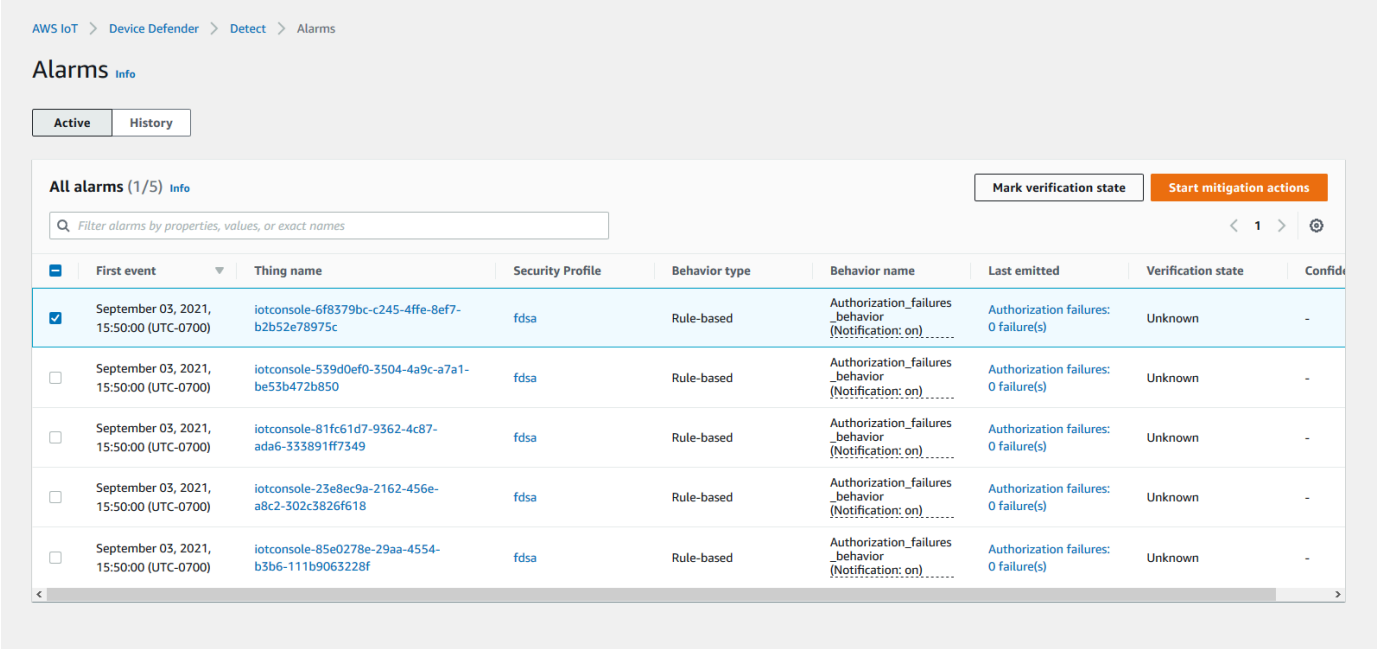
- 启动告警所需的 ML 模型数据点
- 清除告警所需的 ML 模型数据点
- ML Detect 置信级别
- ML Detect 通知，例如 Not suppressed（未隐藏）、Suppressed（隐藏）。



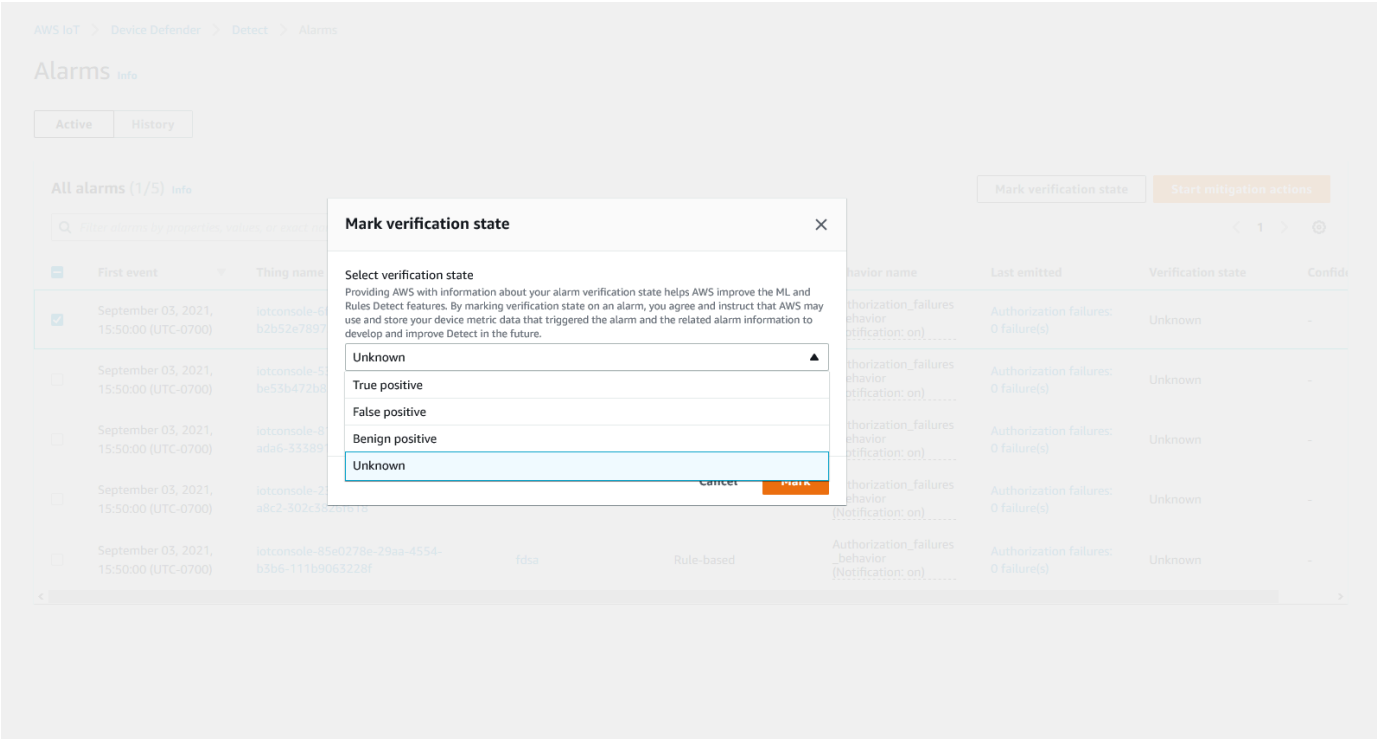
标记告警的验证状态

通过设置验证状态并提供该验证状态的描述来标记告警。这会帮助您和您的团队识别不必回应的告警。

1. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Alarms（告警）。选择告警以标记其验证状态。



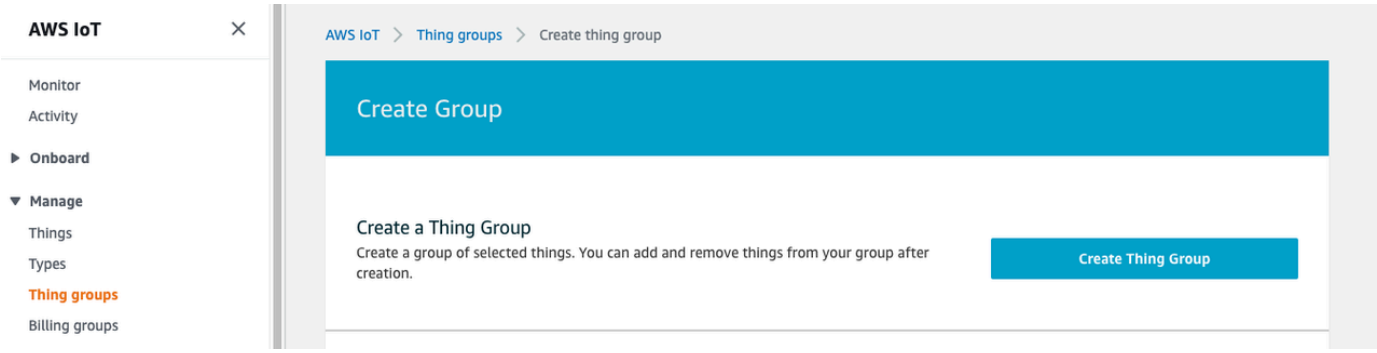
2. 选择 Mark verification state（标记验证状态）。验证状态模式打开。
3. 选择适当的验证状态，输入验证说明（可选），然后选择 Mark（标记）。此操作将验证状态和描述分配给选定的告警。



缓解已确定的设备问题

1. （可选）在设置隔离缓解措施之前，让我们设置一个隔离组，我们会将违规的设备移动到该组。您也可以使用现有组。
2. 导航到 Manage（管理）、Thing groups（事物组），然后选择 Create Thing Group（创建事物组）。为您的事物组命名。在本教程中，我们将事物组命名为 `Quarantine_group`。在 Thing Group（事物组）、Security（安全）中，将以下策略应用于事物组。

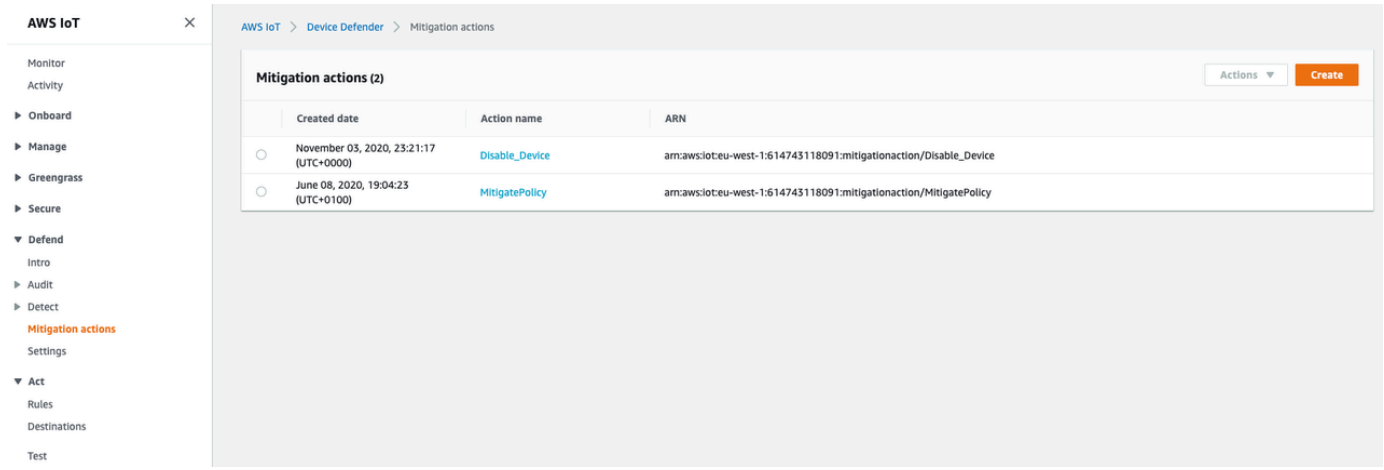
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "iot:*",
      "Resource": "*",
    }
  ]
}
```



完成后，选择 Create thing group（创建事务组）。

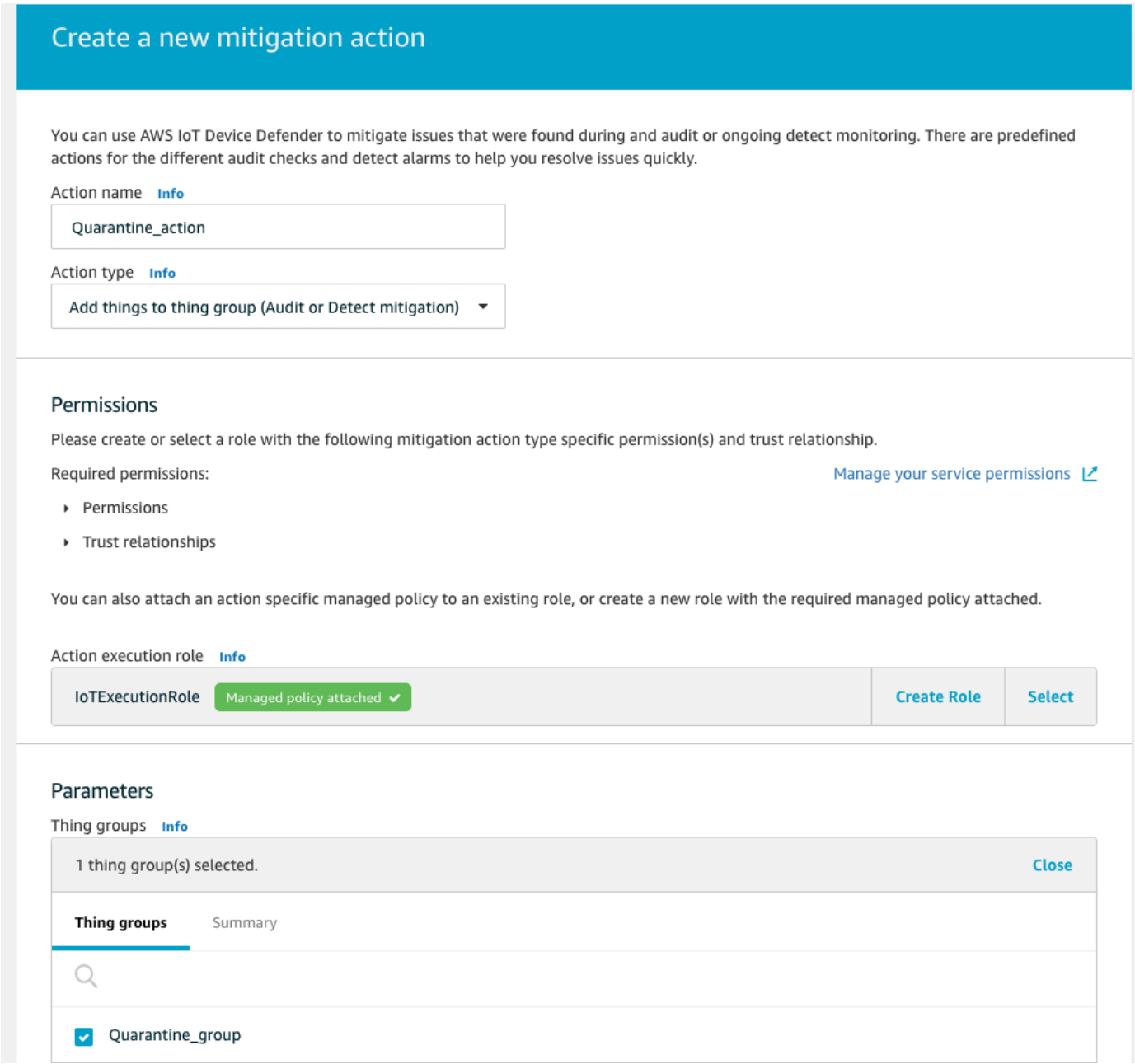
3. 现在我们已经创建了事物组，让我们创建一个缓解操作，将告警中的设备移动到 `Quarantine_group`。

   在 Defend（防护）、Mitigation actions（缓解操作）中，选择 Create（创建）。



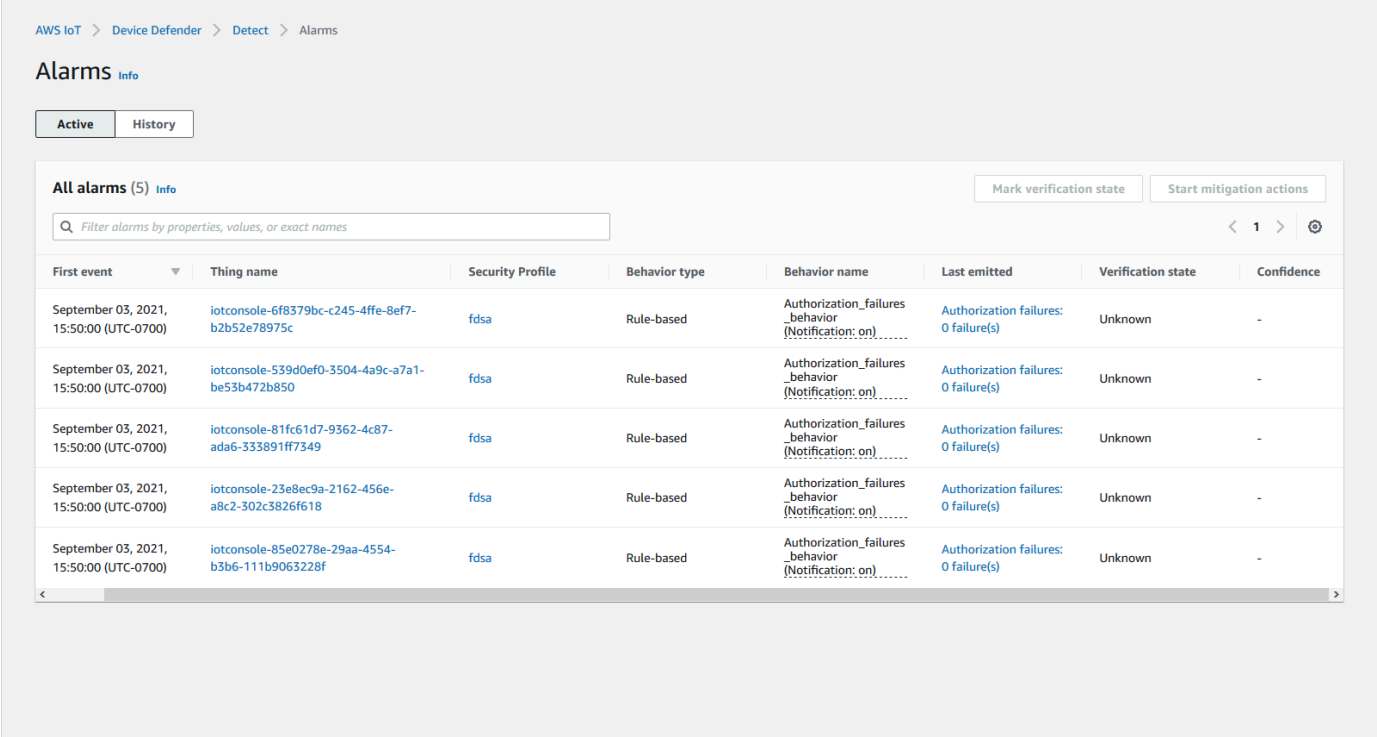
4. 在 Create new mitigation action（创建新的缓解操作）页面上，输入以下信息。

   - Action name（操作名称）：指定缓解操作的名称，例如 **`Quarantine_action`**。
   - Action type（操作类型）：选择操作的类型。我们将选择 Add things to things group (Audit or Detect mitigation)（将事物添加到事物组（审计或 Detect 缓解））。
   - Action execution role（操作执行角色）：创建角色或选择现有角色（如果您之前创建了角色）。
   - Parameters（参数）：选择事物组。我们可以使用之前创建的 `Quarantine_group`。



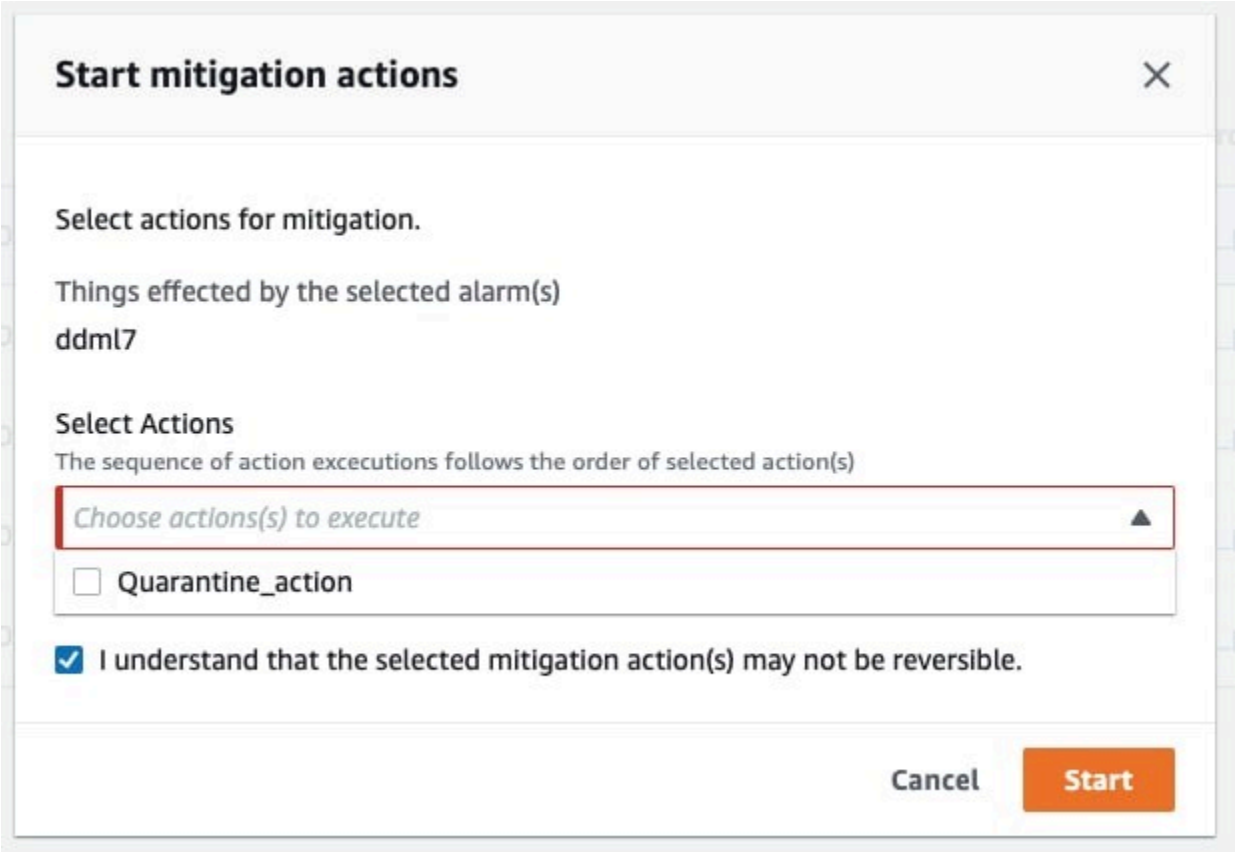
完成此操作后，选择保存。现在，您可以执行缓解操作，将告警中的设备移动到隔离事物组，并在调查期间执行缓解操作来隔离设备。

5. 导航到 Defender、Detect（检测）、Alarms（告警），您可以查看 Active（激活）状态下有哪些设备处于告警状态。

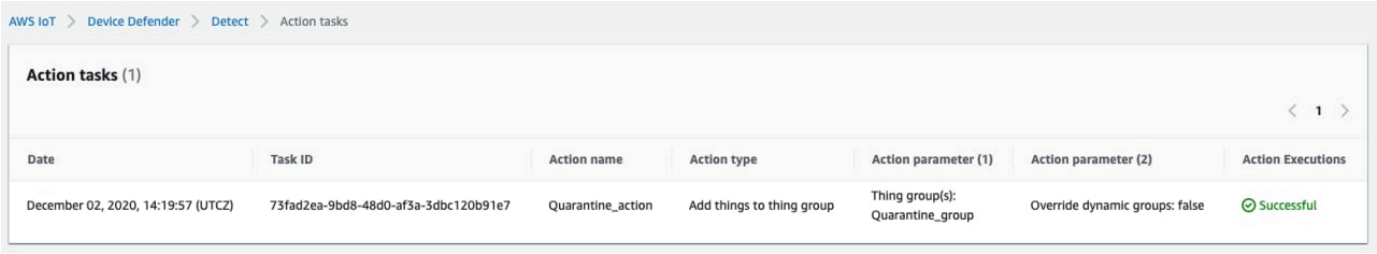


选择要移动到隔离组的设备，然后选择 Start Mitigation Actions（启动缓解操作）。

6. 在 Start mitigation actions（启动缓解操作）、Start Actions（启动操作）项下，选择您之前创建的缓解操作。例如，我们将选择 **`Quarantine_action`**，然后选择 Start（启动）。此时将打开操作任务页面。



7. 设备现在被隔离在 **Quarantine_group** 中，您可以调查引发告警的问题的根本原因。完成调查后，您可以将设备移出事物组或采取进一步操作。



如何将 （ ML Detect 与 CLI 一起使用

下面的介绍了如何使用 CLI 设置 ML Detect。

教程

- 启用 ML Detect
- 监控您的 ML 模型状态
- 查看您的 ML Detect 告警

- 微调您的机器学习（ML）告警
- 标记告警的验证状态
- 缓解已确定的设备问题

启用 ML Detect

以下流程介绍如何在 AWS CLI 中启用 ML Detect。

1. 确保您的设备将如 ML Detect 最低要求中定义创建所需的最小数据点，以进行持续训练和模型更新。为了进行数据收集，请确保您的事物位于附加到安全配置文件的事物组中。
2. 使用 `create-security-profile` 命令创建 ML Detect 安全配置文件。以下示例创建一个名为 *`security-profile-for-smart-lights`* 的安全配置文件，用于检查发送的消息数量、授权失败次数、连接尝试次数以及断开连接次数。此示例使用 `mlDetectionConfig` 以确定该指标将使用 ML Detect 模型。

```
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
  }]'
```

输出：

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

3. 接下来，将您的安全配置文件与一个或多个事物组关联。使用 attach-security-profile 命令将事物组附加到您的安全配置文件。下面的示例将名为 *ML_Detect_beta_static_group* 的事务组关联到了 *security-profile-for-smart-lights* 安全配置文件。

```
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
```

输出：

无。

4. 创建完整的安全配置文件后，ML 模型将开始训练。初始 ML 模型训练和构建需要 14 天才能完成。14 天后，如果您的设备上有异常活动，则可能会看到告警。

监控您的 ML 模型状态

以下流程介绍如何监控正在进行训练的 ML 模型。

- 使用 get-behavior-model-training-summaries 命令查看 ML 模型的进度。以下示例获取 *security-profile-for-smart-lights* 安全配置文件的 ML 模型训练进度摘要。modelStatus 会显示模型是否已完成训练，或是仍在等待针对特定行为的构建。

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

输出：

```
{
    "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**
>
> 如果您的模型未按预期进行，请确保您的设备满足 最低要求 的要求。

查看您的 ML Detect 告警

在构建 ML 模型并准备好进行数据评估之后，您可以定期查看由模型推断的任何告警。以下流程介绍如何在 AWS CLI 中查看您的告警。

- 要查看所有活动告警，请使用 list-active-violations 命令。

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

输出：

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

或者，您可以使用 list-violation-events 命令查看在给定时间内发现的所有违规行为。以下示例列出了从 2020 年 9 月 22 日 5:42:13 GMT 到 2020 年 10 月 26 日 5:42:13 GMT 之间的违规事件。

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

输出：

```
{
    "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"
}
```

微调您的机器学习（ML）告警

构建 ML 模型并准备好进行数据评估后，您可以更新安全配置文件的 ML 行为设置以更改配置。以下流程介绍如何在 AWS CLI 中更新您的安全配置文件的机器学习（ML）行为设置。

- 要更改安全配置文件的 ML 行为设置，请使用 update-security-profile 命令。以下示例通过更改部分行为的 confidenceLevel 并取消隐藏所有行为的通知，从而更新了 *security-profile-for-smart-lights* 安全配置文件的行为。

```
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

  }]'
```

输出：

```
 {
    "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
}
```

标记告警的验证状态

您可以使用验证状态标记告警，帮助对告警进行分类并调查异常情况。

- 用验证状态和状态描述标记告警。例如，要将告警的验证状态设置为误报，请使用以下命令：

```
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
```

输出：

无。

缓解已确定的设备问题

1. 使用 create-thing-group 命令为缓解操作创建事物组。在下面的示例中，我们创建了一个名为 ThingGroupForDetectMitigationAction 的事物组。

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

输出：

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

2. 接下来，使用 create-mitigation-action 命令创建缓解操作。在以下示例中，我们使用了 IAM 角色的 ARN 创建了一个名为 detect_mitigation_action 的缓解操作，该 IAM 角色专用于应用缓解操作。我们还将定义操作类型和该操作的参数。在这种情况下，我们的缓解措施会将事物移动到我们之前创建的名为 ThingGroupForDetectMitigationAction 的事物组。

```
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
     }
 }'
```

输出：

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

3. 使用 start-detect-mitigation-actions-task 命令启动缓解操作任务。task-id、target 和 actions 是必要的参数。

```
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
```

输出：

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

4. （可选）要查看任务中包含的缓解操作执行，请使用 list-detect-mitigation-actions-executions 命令。

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

输出：

```
{
    "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. （可选）使用 describe-detect-mitigation-actions-task 命令获取有关缓解操作任务的信息。

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

输出：

```
{
```

```
    "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. （可选）要获取缓解操作任务的列表，请使用 list-detect-mitigation-actions-tasks 命令。

```
aws iot list-detect-mitigation-actions-tasks \
```

```
    --start-time 1609985315 \
    --end-time 1609988915 \
    --max-items 5 \
    --page-size 4
```

输出：

```
{
    "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. （可选）要取消缓解操作任务，请使用 cancel-detect-mitigation-actions-task 命令。

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

输出：

无。

自定义查看 AWS IoT Device Defender 审计结果的时间和方式

AWS IoT Device Defender 审计提供定期安全检查以确认 AWS IoT 设备和资源都遵循了最佳实践。对于每次检查，审计结果都被归类为合规或不合规，其中不合规结果会导致控制台出现警告图标。为了减少重复已知问题产生的噪音，审计查找结果隐藏特征允许您临时将这些不合规通知静音。

您可以在预定时间段内隐藏指定资源或账户的特定审计检查。已隐藏的审计检查结果将被归类为隐藏的查找结果，与合规类别和不合规类别区分开。此新类别不会触发像不合规结果那样的告警。这样，您就可以在已知维护期间或计划完成更新之前减少不合规通知干扰。

开始使用

以下各节详细介绍了如何使用审计查找结果隐藏功能来隐藏控制台和 CLI 中的 `Device certificate expiring` 检查。如果您想遵循其中一个演示，您必须首先创建两个即将到期的证书，以便 Device Defender 检测。

请使用以下命令创建您的证书。

- 《AWS IoT Core 开发人员指南》中的创建和注册 CA 证书
- 使用您的 CA 证书创建客户端证书 在步骤 3 中，将您的 `days` 参数设置为 **1**。

如果您正在使用 CLI 创建证书，请输入以下命令。

```
openssl x509 -req \
    -in device_cert_csr_filename \
```

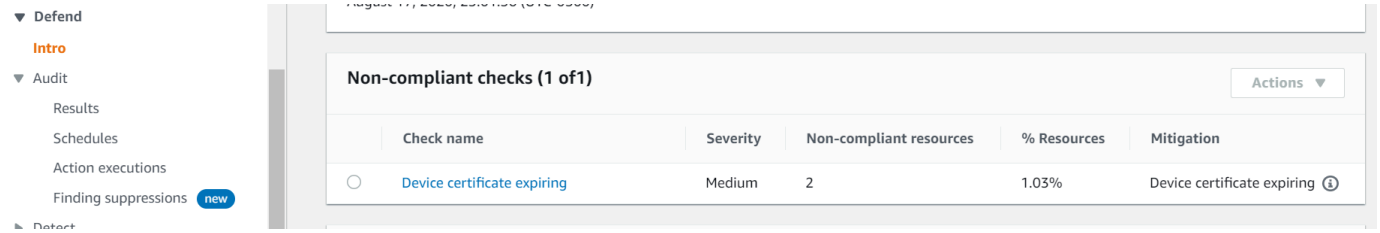
```
    -CA root_ca_pem_filename \
    -CAkey root_ca_key_filename \
    -CAcreateserial \
    -out device_cert_pem_filename \
    -days 1 -sha256
```

在控制台中自定义审计查找结果

以下演练使用具有两个过期设备证书的账户，这些证书将触发不合规审计检查。在这种情况下，我们希望禁用警告，因为我们的开发人员正在测试一项新特征来解决问题。我们为每个证书创建审计查找结果抑制，以阻止下周审计结果出现不合规的情况。

1. 我们将首先运行按需审计，以显示过期的设备证书检查不合规。

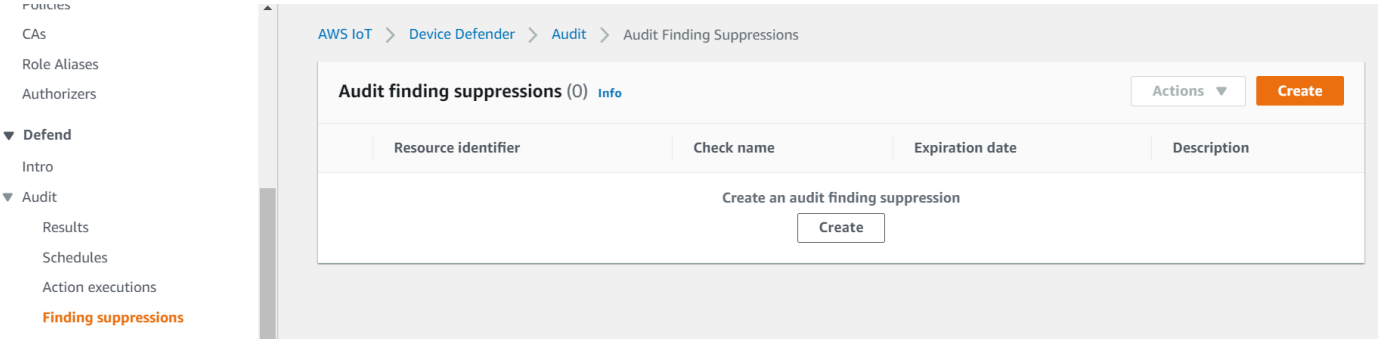
   从 AWS IoT控制台中，从左侧边栏选择 Defend（防护），然后 Audit（审计），随后选择 Results（结果）。在 Audit Results（审计结果）页面上，选择 Create（创建）。Create a new audit（创建新审计）窗口将会打开。选择创建。



   从按需审计结果中，我们可以看到两个资源的“设备证书即将到期”不合规。

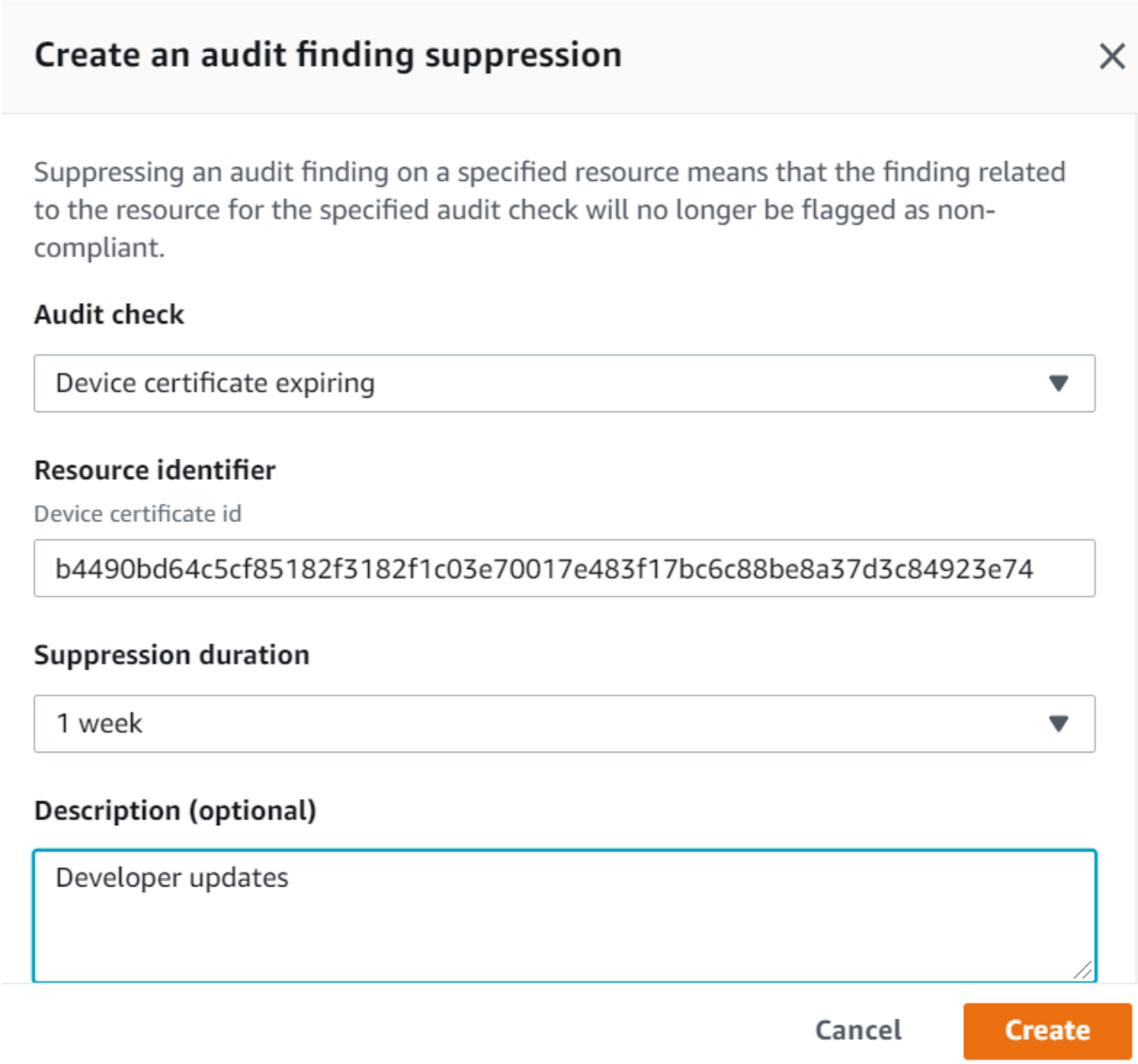
2. 现在，我们希望禁用“设备证书即将过期”的不合规检查警告，因为我们的开发人员正在测试新特征来修复警告。

   从左侧边栏的 Defend（防护）下，选择 Audit（审计），然后选择 Finding suppressions（查找结果隐藏）。在 Audit finding suppressions（审计查找结果隐藏）页面上，选择 Create（创建）。



3. 在 Create an audit finding suppression（创建审计查找结果隐藏）窗口中，我们需要填写以下内容。

- Audit check:（审计检查）：我们选择 `Device certificate expiring`，因为这是我们想要隐藏的审计检查。
- Resource identifier（资源标识符）：我们输入希望隐藏审计查找结果的证书之一的设备证书 ID。
- Suppression duration（隐藏时长）：我们选择 `1 week`，因为这是我们希望将 `Device certificate expiring` 审计检查隐藏的时长。
- Description (optional)（描述（可选））：我们添加了一个说明，描述了为什么我们要隐藏此审计查找结果。

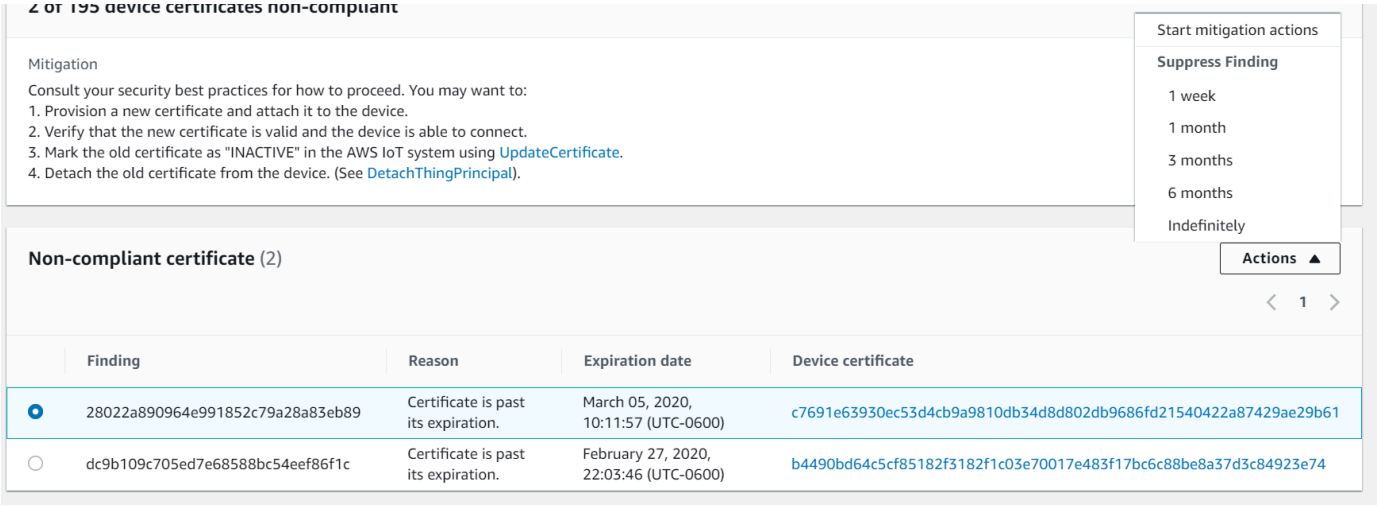


填写完这些字段后，选择 Create（创建）。创建审计查找结果隐藏后，我们会看到一个成功的提示。

4. 我们隐藏了其中一个证书的审计查找结果，现在我们需要隐藏第二个证书的审计查找结果。我们可以使用与步骤 3 中相同的隐藏方法，但此次我们将使用不同的方法进行演示。

从 Defend（防护）的左侧边栏中，选择 Audit（审计），然后选择 Results（结果）。在 Audit results（审计结果）页面上，选择包含不合规资源的审计。然后，在 Non-compliant checks（不合规检查）项下选择资源。在我们的例子中，我们选择“设备证书即将到期”。

5. 在 Device certificate expiring（设备证书即将到期）页面，在 Non-compliant policy（不合规的策略）项下选择需要隐藏的查找结果旁的选项按钮。接下来，选择 Actions（操作）下拉菜单，然后选择为要隐藏的查找结果选择时长。在我们的例子中，我们跟其它证书一样选择 `1 week`。在 Confirm suppression（确认隐藏）窗口中，选择 Enable suppression（启用隐藏）。



创建审计查找结果隐藏后，我们会看到一个成功的提示。现在，两个审计查找结果均已被隐藏 1 周，同时我们的开发人员正在研制解决方案来解决警告。

在 CLI 中自定义您的审计查找结果

以下演练使用具有过期设备证书并触发了不合规审计检查的账户。在这种情况下，我们希望禁用警告，因为我们的开发人员正在测试一项新特征来解决问题。我们为证书创建审计查找结果隐藏，以防下周的审计结果出现不合规的情况。

我们将使用以下 CLI 命令。

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

1. 使用以下命令来启用审计。

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

输出：

无。

2. 使用以下命令运行按需审计，以针对 DEVICE_CERTIFICATE_EXPIRING_CHECK 审计检查。

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

输出：

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

3. 使用 describe-account-audit-configuration 命令来描述审计配置。我们希望确认我们已开启 DEVICE_CERTIFICATE_EXPIRING_CHECK 的审计检查。

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

输出：

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

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

DEVICE_CERTIFICATE_EXPIRING_CHECK 应具有 true 值。

4. 使用 list-audit-task 命令来标识已完成的审计任务。

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

输出：

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

您在步骤 1 中运行的审计的 taskId 应该具有一个为 COMPLETED 的 taskStatus。

5. 使用 describe-audit-task 命令，获取有关使用来自前述步骤的 taskId 输出完成的审计的详细信息。此命令列出了有关审计的详细信息。

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

输出：

```
{
    "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. 使用 list-audit-findings 命令查找不合规的证书 ID，以便我们可以暂停此资源的审计提示。

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

输出：

```
{
    "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. 使用 create-audit-suppression 命令将 ID 为 *c7691e<shortened>* 的设备证书隐藏 DEVICE_CERTIFICATE_EXPIRING_CHECK 审计检查的通知，直至 *2020-08-20*。

```
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. 使用 list-audit-suppression 命令确认审计隐藏设置并获取有关隐藏的详细信息。

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

输出：

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

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

9. update-audit-suppression 命令可用于更新审计查找结果隐藏。下面的示例将 `expiration-date` 更新为了 `08/21/20`。

```
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 命令可用于删除审计查找结果隐藏。

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

要确认删除，请使用 list-audit-suppressions 命令。

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

输出：

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

在本教程中，我们向您展示了如何在控制台和 CLI 中隐藏 `Device certificate expiring` 检查。有关审计查找结果隐藏的更多信息，请参阅 审计查找结果隐藏。

审核

AWS IoT Device Defender 审核会查看与账户相关的和与设备相关的设置及策略，以确保安全措施就绪。审核可帮助您检测与安全最佳实践或访问策略的任何偏离（例如，使用同一身份的多个设备，或允许一个设备读取和更新许多其它设备数据的过度宽松的权限策略。）您可以根据需要运行审核（按需审核）或安排审核定期运行（计划审核）。

AWS IoT Device Defender 审核会针对常见的 IoT 安全最佳实践和设备漏洞运行一组预定义检查。预定义检查示例包括，授权读取或更新多个设备上数据的策略、共享身份（X.509 证书）的设备或者即将过期或已吊销但仍处于活动状态的证书。

问题严重性

问题严重性指示与每个已标识的不合规实例相关的关注级别，以及建议的补救时间。

重大

具有此严重性的不合规审计检查将标识需要迫切关注的问题。关键问题通常使得只掌握了少量高级知识且没有中间人知识或特殊凭证的不良行为者，可以轻松获取您资产的访问或控制权限。

高

具有此严重性的不合规审计检查需要立即进行调查，并在解决关键问题之后提供补救计划。与关键问题一样，高严重性问题通常会向不良行为者提供对您资产的访问或控制权限。但是，高严重性问题通常更难以利用。它们可能需要特殊的工具、中间人知识或特定的设置。

中

具有此严重性的不合规审计检查会提出问题，需要在持续的安保状况维护过程中加以注意。中等严重性问题可能会导致负面操作影响，例如由于安全控制故障而导致的意外中断。这些问题也可能向不良行为者提供对您资产的有限访问或控制，或者可能有助于其部分恶意行为的实施。

低

具有此严重性的不合规审计检查通常表明忽略或绕过了安全最佳实践。虽然它们本身可能不会对安全造成直接影响，但这些失误可能被不良行为者利用。与中等严重性问题一样，低严重性问题需要在持续的安全状况维护过程中加以注意。

后续步骤

要了解可执行的审计检查类型，请参阅 审核检查。有关适用于审计的服务配额的信息，请参阅 Service Quotas。

审核检查

> **Note**
>
> 当您启用检查后，数据收集就会立即开始。如果账户中有大量数据需要收集，那么在启用检查之后可能需要等待一定时间才会获得结果。

以下为受支持的审核检查：

- 已吊销中间 CA 以进行活动设备证书检查
- 已吊销的 CA 证书仍处于活动状态
- 共享设备证书
- 设备证书密钥质量
- CA 证书密钥质量
- 未经身份验证的 Cognito 角色过于宽容
- 经过身份验证的 Cognito 角色过于宽容
- AWS IoT 策略过于宽容
- AWS IoT 策略可能配置错误
- 角色别名过于宽容
- 角色别名允许访问未使用的服务
- CA 证书即将过期
- 冲突的 MQTT 客户端 ID
- 设备证书即将过期
- 设备证书期限检查
- 已撤销的设备证书仍处于激活状态
- 日志记录已禁用

已吊销中间 CA 以进行活动设备证书检查

使用此检查来识别尽管撤消了中间 CA，但仍处于活动状态的所有相关设备证书。

此检查在 CLI 和 API 中显示为 `INTERMEDIATE_CA_REVOKED_FOR_ACTIVE_DEVICE_CERTIFICATES_CHECK`。

严重性：危急

详细信息

此检查发现不合规问题时，会返回以下原因代码：

- INTERMEDIATE_CA_REVOKED_BY_ISSUER

为什么这非常重要

已吊销中间 CA 以进行活动设备证书检查通过确定 CA 链中已吊销中间颁发 CA 的 AWS IoT Core 中是否存在活动设备证书来评估设备身份和信任度。

不应再使用已吊销的中间 CA 来签署 CA 链中的任何其他 CA 或设备证书。在吊销中间 CA 后，具有使用此 CA 证书签名的证书的新添加设备将构成安全威胁。

如何修复

查看 CA 证书被吊销后一段时间的设备证书注册活动。按照您的安全最佳做法来缓解这种情况。您可能需要：

1. 为受影响的设备配置由其他 CA 签名的新证书。
2. 验证新证书是否有效，以及设备能否使用新证书进行连接。
3. 使用 UpdateCertificate 在 AWS IoT 中将旧证书标记为“REVOKED”。您还可以使用缓解操作实现以下目的：
   - 对您的审计查找结果应用 `UPDATE_DEVICE_CERTIFICATE` 缓解操作以进行此更改。
   - 应用 `ADD_THINGS_TO_THING_GROUP` 缓解操作，以将设备添加到可以对其执行操作的组。
   - 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 `PUBLISH_FINDINGS_TO_SNS` 缓解操作。
   - 对吊销中间 CA 证书后的设备证书注册活动进行审核，并考虑吊销在此期间可能使用它颁发的任何设备证书。可以使用 ListRelatedResourcesForAuditFinding 列出通过该 CA 证书签发的设备证书，并使用 UpdateCertificate 吊销设备证书。

- 将旧证书从设备分离。（请参阅 DetachThingPrincipal。）

有关更多信息，请参阅 缓解操作。

已吊销的 CA 证书仍处于活动状态

CA 证书已被吊销，但在 AWS IoT 中仍处于活动状态。

此检查在 CLI 和 API 中显示为 `REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK`。

严重性：危急

详细信息

CA 证书在发证机构所维护的证书吊销列表中标记为已吊销，但在 AWS IoT 中仍标记为“ACTIVE”或“PENDING_TRANSFER”。

此检查发现不合规的 CA 证书时，会返回以下原因代码：

- CERTIFICATE_REVOKED_BY_ISSUER

为什么这非常重要

已吊销的 CA 证书不应再用于签发设备证书。证书可能因遭破坏而被吊销。如果新添加的设备具有使用此 CA 证书签发的证书，可能会造成安全威胁。

如何修复

1. 使用 UpdateCACertificate 在 AWS IoT 中将该 CA 证书标记为“INACTIVE”。您还可以使用缓解操作实现以下目的：
   - 对您的审计查找结果应用 `UPDATE_CA_CERTIFICATE` 缓解操作以进行此更改。
   - 应用 `PUBLISH_FINDINGS_TO_SNS` 缓解操作，以实施自定义响应来响应 Amazon SNS 消息。

   有关更多信息，请参阅 缓解操作。
2. 对吊销 CA 证书后的设备证书注册活动进行审核，并考虑吊销在此期间可能使用它颁发的任何设备证书。可以使用 ListCertificatesByCA 列出通过该 CA 证书签发的设备证书，并使用 UpdateCertificate 吊销设备证书。

共享设备证书

多个并发连接使用相同的 X.509 证书向 AWS IoT 进行身份验证。

此检查在 CLI 和 API 中显示为 `DEVICE_CERTIFICATE_SHARED_CHECK`。

严重性：危急

详细信息

在按需审计过程中进行此检查时，它会检查在审计开始之前 31 天至检查运行前 2 小时内设备用于连接的证书和客户端 ID。对于计划审计，此检查会查看从上次运行审计前 2 小时到该审计实例开始前 2 小时期间的数据。如果在检查期间已采取措施来缓解这种状况，请记录执行并行连接的时间，以判断问题是否持续存在。

此检查发现不合规的证书时，会返回以下原因代码：

- CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES

此外，此检查返回的查找结果还包含共享证书的 ID、使用证书进行连接的客户端的 ID，以及连接/连接断开次数。最近的结果列在最前面。

为什么这非常重要

每个设备应具有唯一证书以向 AWS IoT 进行身份验证。如果多个设备使用相同的证书，这可能表示设备已遭破坏。其身份可能已遭克隆，会进一步危害系统。

如何修复

验证设备证书是否已遭破坏。如果已遭破坏，请遵照安全最佳实践来缓解此情况。

如果在多个设备上使用同一证书，则需要执行以下操作：

1. 预置新的唯一证书并将其附加到每个设备。
2. 验证新证书是否有效，以及设备能否使用它们进行连接。
3. 使用 UpdateCertificate 在 AWS IoT 中将旧证书标记为“REVOKED”。您还可以使用缓解操作实现以下目的：
   - 对您的审计查找结果应用 `UPDATE_DEVICE_CERTIFICATE` 缓解操作以进行此更改。
   - 应用 `ADD_THINGS_TO_THING_GROUP` 缓解操作，以将设备添加到可以对其执行操作的组。

- 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 PUBLISH_FINDINGS_TO_SNS 缓解操作。

有关更多信息，请参阅 缓解操作。

4. 将旧证书从各个设备中分离。

设备证书密钥质量

AWS IoT 客户通常依赖使用 X.509 证书的 TLS 相互身份验证，用于对 AWS IoT 消息代理进行身份验证。这些证书及其证书颁发机构证书必须先在其 AWS IoT 账户中注册，然后才能使用。在注册这些证书时，AWS IoT 对这些证书执行基本的完整性检查。这些检查包括：

- 必须采用有效的格式。
- 必须由注册的证书颁发机构签名。
- 必须仍然在其有效期内（换句话说，尚未过期）。
- 其加密密钥大小必须满足所需的最小大小（对于 RSA 密钥，它们必须为 2048 位或更大）。

此审计检查为您的加密密钥质量提供了以下附加测试：

- CVE-2008-0166 — 检查是否在基于 Debian 的操作系统上，使用 OpenSSL 0.9.8c-1 到 0.9.8g-9 之间的版本生成了密钥。这些版本的 OpenSSL 使用随机数生成器生成可预测的数字，使远程攻击者更容易对加密密钥进行暴力猜测攻击。
- CVE-2017-15361 - 检查密钥是否由 Infineon RSA 库 1.02.013 在 Infineon Trusted Platform Module (TPM) 固件中生成，例如 0000000000000422 - 4.34 之前的版本、000000000000062b - 6.43 之前的版本以及 0000000000008521 - 133.33 之前的版本。该库不当地处理 RSA 密钥生成，使攻击者更容易通过针对性的攻击，破解某些加密保护机制。受影响的技术示例包括使用 TPM 1.2 的 BitLocker、YubiKey 4（4.3.5 之前）PGP 密钥生成以及 Chrome 操作系统中的缓存用户数据加密功能。

如果证书未通过这些测试，AWS IoT Device Defender 会将证书报告为不合规。

此检查在 CLI 和 API 中显示为 DEVICE_CERTIFICATE_KEY_QUALITY_CHECK。

严重性：危急

详细信息

此检查适用于状态为“ACTIVE”或“PENDING_TRANSFER”的设备证书。

此检查发现不合规的证书时，会返回以下原因代码：

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

为什么这非常重要

当设备使用易受攻击的证书时，攻击者会更容易盗用该设备。

如何修复

更新您的设备证书以替换具有已知漏洞的这些证书。

如果在多个设备上使用同一证书，则需要执行以下操作：

1. 预置新的唯一证书并将其附加到每个设备。
2. 验证新证书是否有效，以及设备能否使用它们进行连接。
3. 使用 UpdateCertificate 在 AWS IoT 中将旧证书标记为“REVOKED”。您还可以使用缓解操作实现以下目的：
   - 对您的审计查找结果应用 UPDATE_DEVICE_CERTIFICATE 缓解操作以进行此更改。
   - 应用 ADD_THINGS_TO_THING_GROUP 缓解操作，以将设备添加到可以对其执行操作的组。
   - 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 PUBLISH_FINDINGS_TO_SNS 缓解操作。

   有关更多信息，请参阅 缓解操作。
4. 将旧证书从各个设备中分离。

CA 证书密钥质量

AWS IoT 客户通常依赖使用 X.509 证书的 TLS 相互身份验证，用于对 AWS IoT 消息代理进行身份验证。这些证书及其证书颁发机构证书必须先在其 AWS IoT 账户中注册，然后才能使用。在注册这些证书时，AWS IoT 对这些证书执行基本的完整性检查，包括：

- 证书采用有效格式。

- 证书必须在其有效期内（换句话说，尚未过期）。
- 其加密密钥大小满足所需的最小大小（对于 RSA 密钥，它们必须为 2048 位或更大）。

此审计检查为您的加密密钥质量提供了以下附加测试：

- CVE-2008-0166 — 检查是否在基于 Debian 的操作系统上，使用 OpenSSL 0.9.8c-1 到 0.9.8g-9 之间的版本生成了密钥。这些版本的 OpenSSL 使用随机数生成器生成可预测的数字，使远程攻击者更容易对加密密钥进行暴力猜测攻击。
- CVE-2017-15361 - 检查密钥是否由 Infineon RSA 库 1.02.013 在 Infineon Trusted Platform Module (TPM) 固件中生成，例如 0000000000000422 - 4.34 之前的版本、000000000000062b - 6.43 之前的版本以及 0000000000008521 - 133.33 之前的版本。该库不当地处理 RSA 密钥生成，使攻击者更容易通过针对性的攻击，破解某些加密保护机制。受影响的技术示例包括使用 TPM 1.2 的 BitLocker、YubiKey 4（4.3.5 之前）PGP 密钥生成以及 Chrome 操作系统中的缓存用户数据加密功能。

如果证书未通过这些测试，AWS IoT Device Defender 会将证书报告为不合规。

此检查在 CLI 和 API 中显示为 `CA_CERTIFICATE_KEY_QUALITY_CHECK`。

严重性：危急

详细信息

此检查适用于状态为“ACTIVE”或“PENDING_TRANSFER”的 CA 证书。

此检查发现不合规的证书时，会返回以下原因代码：

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

为什么这非常重要

如果新添加的设备使用此 CA 证书签名，可能会造成安全威胁。

如何修复

1. 使用 UpdateCACertificate 在 AWS IoT 中将该 CA 证书标记为“INACTIVE”。您还可以使用缓解操作实现以下目的：

- 对您的审计查找结果应用 UPDATE_CA_CERTIFICATE 缓解操作以进行此更改。
- 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 PUBLISH_FINDINGS_TO_SNS 缓解操作。

有关更多信息，请参阅 缓解操作。

2. 对吊销 CA 证书后的设备证书注册活动进行审核，并考虑吊销在此期间可能使用它颁发的任何设备证书。（使用 ListCertificatesByCA 列出通过该 CA 证书签发的设备证书，并使用 UpdateCertificate 吊销设备证书。）

未经身份验证的 Cognito 角色过于宽容

附加到未经身份验证的 Amazon Cognito 身份池角色的策略被视为过于宽容，因为它会授权执行以下任意 AWS IoT 操作：

- 管理或修改事物。
- 读取事物管理数据。
- 管理非事物相关的数据或资源。

或者，因为它授权在一系列设备上执行以下 AWS IoT 操作：

- 使用 MQTT 连接、发布或订阅预留主题（包括影子或任务执行数据）。
- 使用 API 命令读取或修改影子或任务执行数据。

一般情况下，使用未经身份验证的 Amazon Cognito 身份池角色进行连接的设备只应具备有限的权限，发布和订阅事物特定 MQTT 主题，或者使用 API 命令读取和修改与影子或任务执行数据相关的事物特定数据。

此检查在 CLI 和 API 中显示为 UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK。

严重性：危急

详细信息

对于此检查，AWS IoT Device Defender 会审计在执行审计之前 31 天内用于连接到 AWS IoT 消息代理的所有 Amazon Cognito 身份池。经过身份验证或未经身份验证的 Amazon Cognito 身份连接的所有 Amazon Cognito 身份池都在审核范围内。

此检查发现不合规的未经身份验证的 Amazon Cognito 身份池角色时，会返回以下原因代码：

- ALLOWS_ACCESS_TO_IOT_ADMIN_ACTIONS
- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

为什么这非常重要

由于未经身份验证的身份从未经过用户的的身份验证，因此它们带来风险的可能性远远高于经过身份验证的 Amazon Cognito 身份。如果未经身份验证的身份遭到破坏，它可能使用管理操作来修改账户设置、删除资源或获取对敏感数据的访问权限。或者，凭借对设备设置宽泛的访问权限，它可以访问或修改您账户中所有设备的影子和任务。来宾用户可能使用这些权限破坏您的整个队列或通过消息发动 DDOS 攻击。

如何修复

附加到未经身份验证的 Amazon Cognito 身份池角色的策略应仅为设备授予执行任务所需的这些权限。我们建议您完成以下步骤：

1. 创建新的合规角色。
2. 创建新的 Amazon Cognito 身份池并为其附加合规角色。
3. 验证您的身份能否使用新池访问 AWS IoT。
4. 验证完成后，将合规角色附加到标记为不合规的 Amazon Cognito 身份池。

您还可以使用缓解操作实现以下目的：

- 应用 `PUBLISH_FINDINGS_TO_SNS` 缓解操作，以实施自定义响应来响应 Amazon SNS 消息。

有关更多信息，请参阅 缓解操作。

管理或修改事物。

以下 AWS IoT API 操作用于管理或修改事物。不应向通过未经身份验证的 Amazon Cognito 身份池进行连接的设备授予执行这些操作的权限。

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`

- DeleteThing
- DetachThingPrincipal
- ListThings
- ListThingsInThingGroup
- RegisterThing
- RemoveThingFromThingGroup
- UpdateThing
- UpdateThingGroupsForThing

任何授权执行这些操作的角色，即便是在单一资源上执行这些操作，也会被视为不合规。

读取事物管理数据

以下 AWS IoT API 操作用于读取或修改事物数据。不应向通过未经身份验证的 Amazon Cognito 的身份池连接的设备授予执行这些操作的权限。

- DescribeThing
- ListJobExecutionsForThing
- ListThingGroupsForThing
- ListThingPrincipals

Example

- 不合规：

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

此示例允许设备执行指定操作，即便该设备仅获得了针对一个事物的权限。

管理非事物对象

通过未经身份验证的 Amazon Cognito 身份池进行连接的设备，不应该被授予执行除这些部分中所述之外的 AWS IoT API 操作的权限。可以通过创建一个不供设备使用的单独身份池，使用通过未经身份验证的 Amazon Cognito 身份池进行连接的应用程序来管理账户。

订阅/发布到 MQTT 主题

MQTT 消息通过 AWS IoT 消息代理发送，并由设备用来执行多个操作，包括访问和修改影子状态和任务执行状态。为设备授权以连接、发布或订阅 MQTT 消息的策略，应该限定对特定资源执行如下操作：

Connect

- 不合规：

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

  通配符 * 允许任何设备连接到 AWS IoT。

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

  除非 `iot:Connection.Thing.IsAttached` 在条件键中设置为 True，这相当于上例中的通配符 *。

- 合规：

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

资源规范包含与用于连接的设备名称匹配的变量。条件语句通过检查 MQTT 客户端所用的证书是否与附加到使用此名称的事物的证书匹配，进一步限制权限。

发布

- 不合规：

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

此示例允许设备更新任何设备的影子（* = 所有设备）。

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

此示例允许设备读取、更新或删除任何设备的影子。

- 合规：

```
{
  "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/*"
      ],
    }
  ]
}
```

资源规范包含通配符，但仅匹配其事物名称用于连接的设备的任何影子相关主题。

Subscribe

- 不合规：

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

此示例允许设备订阅为所有设备预留的影子或任务主题。

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

与上一个示例相同，不过使用的是 # 通配符。

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

此示例允许设备查看任何设备上的影子更新（+ = 所有设备）。

- 合规：

```
{
  "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/*"
      ],
    }
  ]
}
```

资源规范包含通配符，但仅匹配其事物名称用于连接的设备的任何影子相关主题和任务相关主题。

接收

- 合规：

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

这是允许的，因为设备只能从其有权订阅的主题接收消息。

读取/修改影子或任务数据

为设备授予执行 API 操作权限以访问或修改设备影子或任务执行数据的策略，应该限定对特定资源执行以下操作。下面是一些 API 操作：

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

Example

- 不合规：

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

此示例允许设备对任何事物执行指定操作。

- 合规：

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

此示例允许设备仅对两个事物执行指定操作。

经过身份验证的 Cognito 角色过于宽容

附加到经身份验证的 Amazon Cognito 身份池角色的策略被视为过于宽容，因为它会授权执行以下 AWS IoT 操作：

- 管理或修改事物。
- 管理非事物相关的数据或资源。

或者，因为它授权在一系列设备上执行以下 AWS IoT 操作：

- 读取事物管理数据。
- 使用 MQTT 连接/发布/订阅预留主题（包括影子或任务执行数据）。
- 使用 API 命令读取或修改影子或任务执行数据。

一般情况下，使用经身份验证的 Amazon Cognito 身份池角色进行连接的设备只应具备有限的权限，读取事物特定管理数据、发布和订阅事物特定 MQTT 主题，或使用 API 命令读取和修改与影子或任务执行数据相关的事物特定数据。

此检查在 CLI 和 API 中显示为 `AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`。

严重性：危急

详细信息

对于此检查，AWS IoT Device Defender 会审计在执行审计之前 31 天内用于连接到 AWS IoT 消息代理的所有 Amazon Cognito 身份池。经过身份验证或未经身份验证的 Amazon Cognito 身份连接的所有 Amazon Cognito 身份池都在审核范围内。

此检查发现不合规的经身份验证的 Amazon Cognito 身份池角色时，会返回以下原因代码：

- 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

为什么这非常重要

如果经身份验证的身份遭到破坏，它可能使用管理操作来修改账户设置、删除资源或获取对敏感数据的访问权限。

如何修复

附加到经过身份验证的 Amazon Cognito 身份池角色的策略应仅为设备授予执行任务所需的这些权限。我们建议您完成以下步骤：

1. 创建新的合规角色。
2. 创建新的 Amazon Cognito 身份池并为其附加合规角色。
3. 验证您的身份能否使用新池访问 AWS IoT。
4. 验证完成后，将合规角色附加到标记为不合规的 Amazon Cognito 身份池。

您还可以使用缓解操作实现以下目的：

- 应用 PUBLISH_FINDINGS_TO_SNS 缓解操作，以实施自定义响应来响应 Amazon SNS 消息。

有关更多信息，请参阅 缓解操作。

管理或修改事物。

以下 AWS IoT API 操作可用于管理或修改事物，因此不应该为通过经身份验证的 Amazon Cognito 身份池进行连接的设备授予执行以下操作的权限：

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

- DetachThingPrincipal
- ListThings
- ListThingsInThingGroup
- RegisterThing
- RemoveThingFromThingGroup
- UpdateThing
- UpdateThingGroupsForThing

任何授权执行这些操作的角色，即便是在单一资源上执行这些操作，也会被视为不合规。

管理非事物对象

通过经身份验证的 Amazon Cognito 身份池进行连接的设备，不应该被授予执行除这些部分中所述之外的 AWS IoT API 操作的权限。要使用通过经身份验证 Amazon Cognito 身份池进行连接的应用程序来管理账户，请创建一个不供设备使用的单独身份池。

读取事物管理数据

以下 AWS IoT API 操作可用于读取事物数据，因此应该为通过经身份验证的 Amazon Cognito 身份池进行连接的设备授予仅对有限事物集执行以下操作的权限：

- DescribeThing
- ListJobExecutionsForThing
- ListThingGroupsForThing
- ListThingPrincipals

- 不合规：

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

此示例允许设备对任何事物执行指定操作。

- 合规：

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

此示例允许设备仅对一个事物执行指定操作。

- 合规：

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

此示例之所以合规，是因为尽管该资源是使用通配符 (*) 指定的，但它的前面是特定字符串，这将仅限名称包含给定前缀的设备访问事物集。

- 不合规：

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

此示例允许设备对任何事物执行指定操作。

- 合规：

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

此示例允许设备仅对一个事物执行指定操作。

- 合规：

```
{
  "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*"
      ]
```

```
    }
  ]
}
```

此示例之所以合规，是因为尽管该资源是使用通配符 (*) 指定的，但它的前面是特定字符串，这将仅限名称包含给定前缀的设备访问事物集。

订阅/发布到 MQTT 主题

MQTT 消息通过 AWS IoT 消息代理发送，并由设备用来执行许多不同的操作，包括访问和修改影子状态和任务执行状态。为设备授权以连接、发布或订阅 MQTT 消息的策略，应该限定对特定资源执行如下操作：

Connect

- 不合规：

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

  通配符 * 允许任何设备连接到 AWS IoT。

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

  除非 `iot:Connection.Thing.IsAttached` 在条件键中设置为 True，这相当于上例中的通配符 *。

- 合规：

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

```
  ]
}
```

资源规范包含与用于连接的设备名称匹配的变量，且条件语句通过检查 MQTT 客户端所用的证书是否与附加到使用此名称的事物的证书匹配，进一步限制权限。

发布

- 不合规：

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

此示例允许设备更新任何设备的影子（* = 所有设备）。

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

此示例允许设备读取/更新/删除任何设备的影子。

- 合规：

```
{
  "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/*"
      ],
    }
  ]
}
```

资源规范包含通配符，但仅匹配其事物名称用于连接的设备的任何影子相关主题。

Subscribe

- 不合规：

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

此示例允许设备订阅为所有设备预留的影子或任务主题。

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

与上一个示例相同，不过使用的是 # 通配符。

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

此示例允许设备查看任何设备上的影子更新（+ = 所有设备）。

- 合规：

```
{
  "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/*"
      ],
    }
  ]
}
```

资源规范包含通配符，但仅匹配其事物名称用于连接的设备的任何影子相关主题和任务相关主题。

接收

- 合规：

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

这是合规的，因为设备只能从其有权订阅的主题接收消息。

读取或修改影子或任务数据

为设备授予执行 API 操作权限以访问或修改设备影子或任务执行数据的策略，应该限定对特定资源执行以下操作。下面是一些 API 操作：

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

示例

- 不合规：

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

此示例允许设备对任何事物执行指定操作。

- 合规：

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

此示例允许设备仅对两个事物执行指定操作。

AWS IoT 策略过于宽容

AWS IoT 策略提供的权限过于宽泛或不受限制。它为一系列设备授予发送或接收 MQTT 消息的权限，或为一系列设备授予访问或修改影子和任务执行数据的权限。

一般而言，针对某个设备的策略应该授予仅与该设备关联的资源的访问权限，而不应该牵涉其它设备，或只牵涉少量其他设备。除了某些例外情况，在此类策略中使用通配符（例如“*”）指定资源被视为过于宽泛或不受限制。

此检查在 CLI 和 API 中显示为 `IOT_POLICY_OVERLY_PERMISSIVE_CHECK`。

严重性：危急

详细信息

此检查发现不合规的 AWS IoT 策略时，会返回以下原因代码：

- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

为什么这非常重要

策略过于宽容的证书、Amazon Cognito 身份或事物组一旦遭到破坏，就会影响您的整个账户。攻击者可以使用此类宽泛的访问权限，读取或修改所有设备的影子、任务或任务执行数据。或者，攻击者可以使用已遭破坏的证书连接您网络中的恶意设备或发动 DDOS 攻击。

如何修复

按照以下步骤来修复附加到事物、事物组或其它实体的任何不合规策略：

1. 使用 CreatePolicyVersion 创建新的兼容版本的策略。将 `setAsDefault` 标记设置为 True。（这可使此新版本适用于使用策略的所有实体。）

2. 使用 ListTargetsForPolicy 获取策略附加到的目标列表（证书、事物组），并确定组中包含的设备或使用证书进行连接的设备。
3. 验证所有关联的设备能否连接到 AWS IoT。如果设备无法连接，使用 SetPolicyVersion 将默认策略回滚到之前的版本，修改策略，然后重试。

您可以使用缓解操作实现以下目的：

- 对您的审计查找结果应用 REPLACE_DEFAULT_POLICY_VERSION 缓解操作以进行此更改。
- 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 PUBLISH_FINDINGS_TO_SNS 缓解操作。

有关更多信息，请参阅 缓解操作。

使用 AWS IoT Core 策略变量在策略中动态引用 AWS IoT 资源。

MQTT 权限

MQTT 消息通过 AWS IoT 消息代理发送，并由设备用来执行多个操作，包括访问和修改影子状态和任务执行状态。为设备授权以连接、发布或订阅 MQTT 消息的策略，应该限定对特定资源执行如下操作：

Connect

- 不合规：

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

  通配符 * 允许任何设备连接到 AWS IoT。

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

  除非 iot:Connection.Thing.IsAttached 在条件键中设置为 True，这相当于上例中的通配符 *。

- 合规：

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

资源规范包含与用于连接的设备名称匹配的变量。条件语句通过检查 MQTT 客户端所用的证书是否与附加到使用此名称的事物的证书匹配，进一步限制权限。

发布

- 不合规：

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

此示例允许设备更新任何设备的影子（* = 所有设备）。

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

此示例允许设备读取、更新或删除任何设备的影子。

- 合规：

```
{
  "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/*"
      ],
    }
  ]
}
```

资源规范包含通配符，但仅匹配其事物名称用于连接的设备的任何影子相关主题。

Subscribe

- 不合规：

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

此示例允许设备订阅为所有设备预留的影子或任务主题。

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

与上一个示例相同，不过使用的是 # 通配符。

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

此示例允许设备查看任何设备上的影子更新（+ = 所有设备）。

- 合规：

```
{
  "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/*"
      ],
    }
  ]
}
```

资源规范包含通配符，但仅匹配其事物名称用于连接的设备的任何影子相关主题和任务相关主题。

接收

- 合规：

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

这是合规的，因为设备只能从其有权订阅的主题接收消息。

影子和任务权限

为设备授予执行 API 操作权限以访问或修改设备影子或任务执行数据的策略，应该限定对特定资源执行以下操作。下面是一些 API 操作：

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

示例

- 不合规：

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

此示例允许设备对任何事物执行指定操作。

- 合规：

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

此示例允许设备仅对两个事物执行指定操作。

AWS IoT 策略可能配置错误

已确定某项 AWS IoT 策略可能配置错误。错误配置的策略，包括过于宽松的策略，可能会导致安全事件，例如允许设备访问意外资源。

可能配置错误的 AWS IoT 策略检查是一种警告，提醒您确保在更新策略之前仅允许预期的操作。

在此 CLI 和 API 中，该检查显示为 `IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK`。

严重性：中

详细信息

当此检查发现可能配置错误的 AWS IoT 策略时，AWS IoT 会返回以下原因代码：

- POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT
- TOPIC_FILTERS_INTENDED_TO_DENY_ALLOWED_USING_WILDCARDS

为什么这非常重要

错误配置的策略可能会向设备提供超出所需的权限，从而导致意外后果。我们建议仔细考虑该政策，以限制对资源的访问并防止安全威胁。

策略在拒绝语句示例中包含 MQTT 通配符

可能配置错误的 AWS IoT 策略检查会检查 deny 语句中是否有 MQTT 通配符（+ 或 #）。通配符被 AWS IoT 策略视为文字字符串，可能会使策略过于宽松。

以下示例旨在通过在策略中使用 MQTT 通配符 # 来拒绝订阅与 building/control_room 相关的主题。但是，MQTT 通配符在 AWS IoT 策略中没有通配符含义，设备可以订阅 building/control_room/data1。

可能配置错误的 AWS IoT 策略检查将使用原因代码 POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT 标记此策略。

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

以下是正确配置的策略的示例。设备无权订阅 building/control_room/ 的子主题，也无权接收来自 building/control_room/ 的子主题的消息。

```
{
"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/*"
    }
  ]
}
```

旨在拒绝使用通配符允许的主题筛选条件示例

以下示例策略旨在通过拒绝资源 building/control_room/* 来拒绝订阅与 building/control_room 相关的主题。但是，设备可以发送 building/# 订阅请求，并接收来自与 building 相关的所有主题的消息，包括 building/control_room/data1。

可能配置错误的 AWS IoT 策略检查将使用原因代码 TOPIC_FILTERS_INTENDED_TO_DENY_ALLOWED_USING_WILDCARDS 标记此策略。

以下示例策略有权接收关于 building/control_room topics 的消息：

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

```
    }
  ]
}
```

以下是正确配置的策略的示例。设备无权订阅 building/control_room/ 的子主题，也无权接收来自 building/control_room/ 的子主题的消息。

```
{
"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**
>
> 此检查可能会误报。我们建议您评估所有已标记的策略，并使用审计抑制功能标记误报资源。

如何修复

此检查会标记可能配置错误的策略，因此可能会出现误报。使用审计抑制功能标记所有误报，这样以后就不会再标记误报。

您可以按照以下步骤来修复附加到事物、事物组或其他实体的任何不合规策略：

1. 使用 CreatePolicyVersion 创建新的兼容版本的策略。将 `setAsDefault` 标记设置为 True。（这可使此新版本适用于使用策略的所有实体。）

   有关为常见用例创建 AWS IoT 策略的示例，请参阅 AWS IoT Core 开发者指南中的发布/订阅策略示例。
2. 验证所有关联的设备能否连接到 AWS IoT。如果设备无法连接，使用 SetPolicyVersion 将默认策略回滚到之前的版本，修改策略，然后重试。

您可以使用缓解操作实现以下目的：

- 对您的审计查找结果应用 `REPLACE_DEFAULT_POLICY_VERSION` 缓解操作以进行此更改。
- 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 `PUBLISH_FINDINGS_TO_SNS` 缓解操作。

有关更多信息，请参阅 缓解操作。

使用《AWS IoT Core 开发人员指南》中的 IoT Core 策略变量在策略中动态引用 AWS IoT 资源。

角色别名过于宽容

AWS IoT 角色别名提供了一种机制，让连接的设备使用 X.509 证书对 AWS IoT 进行身份验证，然后从与 AWS IoT 角色别名关联的 IAM 角色获取短期 AWS 凭证。必须使用带有身份验证上下文变量的访问策略缩小这些凭证的权限范围。如果您的策略配置不当，您可能会使自己暴露在权限升级攻击中。此审计检查确保 AWS IoT 角色别名提供的临时凭证不会过于宽松。

如果发现以下其中一种条件，将触发此检查：

- 该策略针对此角色别名（例如，“iot:*”、“dynamodb:*”、“iam:*” 等）在过去一年中使用的任何服务提供管理权限。
- 该策略提供对事物元数据操作的广泛访问权限、对受限 AWS IoT 操作的访问权限或对 AWS IoT 数据层面操作的广泛访问权限。
- 该策略提供对“iam”、“cloudtrail”、“guardduty”、“inspector”或“trustedadvisor”等安全审计服务的访问权限。

此检查在 CLI 和 API 中显示为 `IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK`。

严重性：危急

详细信息

此检查发现不合规的 IoT 策略时，会返回以下原因代码：

- 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

为什么这非常重要

通过将权限限制为设备执行其正常操作所需的权限，您可以降低设备受到威胁时账户的风险。

如何修复

按照以下步骤来修复附加到事物、事物组或其它实体的任何不合规策略：

1. 按照使用 AWS IoT Core 凭证提供商授予直接调用 AWS 服务的权限中的步骤对您的角色别名应用限制更严的策略。

您可以使用缓解操作实现以下目的：

- 如果要实现自定义操作以响应 Amazon SNS 消息，请应用 `PUBLISH_FINDINGS_TO_SNS` 缓解操作。

有关更多信息，请参阅 缓解操作。

角色别名允许访问未使用的服务

AWS IoT 角色别名提供了一种机制，让连接的设备使用 X.509 证书对 AWS IoT 进行身份验证，然后从与 AWS IoT 角色别名关联的 IAM 角色获取短期 AWS 凭证。必须使用带有身份验证上下文变量的访问策略缩小这些凭证的权限范围。如果您的策略配置不当，您可能会使自己暴露在权限升级攻击中。此审计检查确保 AWS IoT 角色别名提供的临时凭证不会过于宽松。

如果角色别名有权访问过去一年未用于 AWS IoT 设备的服务，则会触发此检查。例如，如果您的 IAM 角色链接到过去一年中仅使用了 AWS IoT 的角色别名，但附加到角色的策略也授予了对 "iam:getRole" 和 "dynamodb:PutItem" 的权限，则审计报告问题。

此检查在 CLI 和 API 中显示为 IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK。

严重性：中

详细信息

此检查发现不合规的 AWS IoT 策略时，会返回以下原因代码：

- ALLOWS_ACCESS_TO_UNUSED_SERVICES

为什么这非常重要

通过将权限限制为设备执行其正常操作所需的服务，您可以降低设备受到威胁时账户的风险。

如何修复

按照以下步骤来修复附加到事物、事物组或其它实体的任何不合规策略：

1. 按照使用 AWS IoT Core 凭证提供商授予直接调用 AWS 服务的权限中的步骤对您的角色别名应用限制更严的策略。

您可以使用缓解操作实现以下目的：

- 如果要实现自定义操作以响应 Amazon SNS 消息，请应用 PUBLISH_FINDINGS_TO_SNS 缓解操作。

有关更多信息，请参阅 缓解操作。

CA 证书即将过期

CA 证书将在 30 天内过期或已经过期。

此检查在 CLI 和 API 中显示为 CA_CERTIFICATE_EXPIRING_CHECK。

严重性：中

详细信息

此检查适用于状态为“ACTIVE”或“PENDING_TRANSFER”的 CA 证书。

此检查发现不合规的 CA 证书时，会返回以下原因代码：

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

为什么这非常重要

已过期的 CA 证书不应被用于签发新设备证书。

如何修复

有关如何继续，请参阅安全最佳实践。您可能需要：

1. 向 AWS IoT 注册新的 CA 证书。
2. 验证能否使用新的 CA 证书签发设备证书。
3. 使用 UpdateCACertificate 在 AWS IoT 中将旧的 CA 证书标记为“INACTIVE”。您还可以使用缓解操作实现以下目的：
   - 对您的审计查找结果应用 UPDATE_CA_CERTIFICATE 缓解操作以进行此更改。
   - 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 PUBLISH_FINDINGS_TO_SNS 缓解操作。

   有关更多信息，请参阅 缓解操作。

冲突的 MQTT 客户端 ID

多个设备使用同一客户端 ID 连接。

此检查在 CLI 和 API 中显示为 CONFLICTING_CLIENT_IDS_CHECK。

严重性：高

详细信息

使用同一客户端 ID 建立了多个连接，从而导致已连接的设备断开连接。MQTT 规范只允许每个客户端 ID 有一个活动连接，因此当另一个设备使用同一客户端 ID 连接时，它会使前一个连接断开。

在按需审核过程中进行此检查时，它会检查在审核开始之前 31 天内客户端 ID 是如何用于连接的。对于计划审核，此检查会查看从上次运行审核到该审核实例开始期间的数据。如果您已在检查期间采取措施来缓解这种状况，请记录连接/连接断开的时间，以判断问题是否持续存在。

此检查发现不合规问题时，会返回以下原因代码：

- DUPLICATE_CLIENT_ID_ACROSS_CONNECTIONS

此检查返回的查找结果还包含用于连接的客户端 ID、委托人 ID 和连接断开次数。最近的结果列在最前面。

为什么这非常重要

ID 相冲突的设备将被迫不断重新连接，这可能导致消息丢失或致使设备无法连接。

这可能表示设备或设备的凭证已遭破坏，并可能是 DDoS 攻击的一部分。也有可能是设备未在账户中得到正确配置，或者设备连接效果不佳，被迫每分钟重新连接多次。

如何修复

将每个设备注册为 AWS IoT 中的唯一事物，并使用事物名称作为客户端 ID 进行连接。或者，在通过 MQTT 连接设备时使用 UUID 作为客户端 ID。您还可以使用缓解操作实现以下目的：

- 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 PUBLISH_FINDINGS_TO_SNS 缓解操作。

有关更多信息，请参阅 缓解操作。

设备证书即将过期

设备证书将在配置的阈值期限内到期或已到期。证书到期检查阈值可以配置在 30 天（最短）到 3652 天（10 年，最长）之间，默认值为 30 天。

此检查在 CLI 和 API 中显示为 DEVICE_CERTIFICATE_EXPIRING_CHECK。

严重性：中

详细信息

此检查适用于状态为“ACTIVE”或“PENDING_TRANSFER”的设备证书。

此检查发现不合规的设备证书时，会返回以下原因代码：

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

为什么这非常重要

设备证书过期后不应再投入使用。

配置设备证书到期检查

此配置使您能够监控设备队列中即将到期的证书和接收其提醒。例如，如果您想在证书到期的 30 天内收到通知，则可以按以下方式配置检查：

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

如何修复

有关如何继续，请参阅安全最佳实践。您可能需要：

1. 预置新证书并将它附加到设备。
2. 验证新证书是否有效以及设备能否使用它进行连接。
3. 使用 UpdateCertificate 在 AWS IoT 中将旧证书标记为“INACTIVE”。您还可以使用缓解操作实现以下目的：
   - 对您的审计查找结果应用 UPDATE_DEVICE_CERTIFICATE 缓解操作以进行此更改。
   - 应用 ADD_THINGS_TO_THING_GROUP 缓解操作，以将设备添加到可以对其执行操作的组。
   - 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 PUBLISH_FINDINGS_TO_SNS 缓解操作。

有关更多信息，请参阅 缓解操作。

4. 将旧证书从设备分离。（请参阅 DetachThingPrincipal。）

设备证书期限检查

当设备证书处于活动状态的天数大于或等于您指定的天数时，此审计检查会提醒您。此检查有助于您随时了解证书的状态，无论证书何时达到其使用寿命的终点，都能定期及时采取行动，通过降低证书泄露的风险来提高安全性。

证书期限检查阈值可以配置在 30 天（最短）到 3652 天（10 年，最长）之间，默认值为 365 天。

此检查在 CLI 和 API 中显示为 DEVICE_CERTIFICATE_AGE_CHECK。默认情况下，此检查处于禁用状态。严重性：低

详细信息

此检查适用于状态为“ACTIVE”或“PENDING_TRANSFER”的设备证书。此检查发现不合规的设备证书时，会返回以下原因代码：

- CERTIFICATE_PAST_AGE_THRESHOLD

配置设备证书期限检查

此配置可让您根据设备队列的特定需求定制证书轮换提醒，有助于您在所有设备上保持强大的安全态势。您可以使用 UpdateAccountAuditConfiguration API 配置此检查。例如，如果您希望在证书处于活动状态超过 365 天时收到提醒，可以按如下方式配置检查：

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

已撤销的设备证书仍处于激活状态

已吊销的设备证书仍处于活动状态。

此检查在 CLI 和 API 中显示为 `REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK`。

严重性：中

详细信息

设备证书位于其 CA 的证书吊销列表中，但它在 AWS IoT 中仍处于活动状态。

此检查适用于状态为“ACTIVE”或“PENDING_TRANSFER”的设备证书。

此检查发现不合规问题时，会返回以下原因代码：

- CERTIFICATE_REVOKED_BY_ISSUER

为什么这非常重要

设备证书通常因为遭到破坏而被吊销。证书也可能因为错误或疏忽而尚未在 AWS IoT 中吊销。

如何修复

验证设备证书是否已遭破坏。如果已遭破坏，请遵照安全最佳实践来缓解此情况。您可能需要：

1. 为设备预置新证书。
2. 验证新证书是否有效以及设备能否使用它进行连接。
3. 使用 UpdateCertificate 在 AWS IoT 中将旧证书标记为“REVOKED”。您还可以使用缓解操作实现以下目的：
   - 对您的审计查找结果应用 `UPDATE_DEVICE_CERTIFICATE` 缓解操作以进行此更改。
   - 应用 `ADD_THINGS_TO_THING_GROUP` 缓解操作，以将设备添加到可以对其执行操作的组。
   - 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 `PUBLISH_FINDINGS_TO_SNS` 缓解操作。

   有关更多信息，请参阅 缓解操作。
4. 将旧证书从设备分离。（请参阅 DetachThingPrincipal。）

日志记录已禁用

Amazon CloudWatch 中未启用 AWS IoT 日志。验证 V1 和 V2 日志记录。

此检查在 CLI 和 API 中显示为 `LOGGING_DISABLED_CHECK`。

严重性：低

详细信息

此检查发现不合规问题时，会返回以下原因代码：

- LOGGING_DISABLED

为什么这非常重要

借助 CloudWatch 中的 AWS IoT 日志，您可以了解 AWS IoT 中的行为，包括身份验证失败、意外的连接和连接断开，这些行为可能表明设备遭到破坏。

如何修复

在 CloudWatch 中启用 AWS IoT 日志。请参阅《AWS IoT Core 开发人员指南》中的日志记录和监控。您还可以使用缓解操作实现以下目的：

- 对您的审计查找结果应用 `ENABLE_IOT_LOGGING` 缓解操作以进行此更改。
- 如果要实现自定义响应以响应 Amazon SNS 消息，请应用 `PUBLISH_FINDINGS_TO_SNS` 缓解操作。

有关更多信息，请参阅 缓解操作。

审核命令

管理审核设置

使用`UpdateAccountAuditConfiguration` 为账户配置审核设置。通过此命令，您可以启用希望用于审核、设置可选通知及配置权限的检查。

使用 `DescribeAccountAuditConfiguration` 检查这些设置。

使用 DeleteAccountAuditConfiguration 删除审核设置。这会还原所有默认值，并有效地禁用审核，因为所有检查默认处于禁用状态。

UpdateAccountAuditConfiguration

针对此账户配置或重新配置 Device Defender 审核设置。设置包括审核通知的发送方式以及启用或禁用的审核检查类型。

摘要

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

cli-input-json 格式

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

cli-input-json 字段

名称	类型	描述
roleArn	字符串	角色的 ARN，该角色授权 AWS IoT 在执行审核时访问设

名称	类型	描述
	最大长度：2048，最小长度：20	备、策略、证书及其它项相关信息。
auditNotificationTargetConfigurations	映射	审核通知要发送到的目标的相关信息。
targetArn	字符串	审核通知要发送到的目标（SNS 主题）的 ARN。
roleArn	字符串 最大长度：2048，最小长度：20	可授权向目标发送通知的角色的 ARN。
已启用	布尔值	如果已启用向目标发送通知，则为 true。

名称	类型	描述
auditCheckConfigurations	映射	为此账户指定要启用和禁用的审核检查。使用 `DescribeAccountAuditConfiguration` 查看所有检查的列表，包括当前已启用的检查。 某些检查在启用后，可能会立即启动某些数据收集。如果禁用某项检查，则将删除迄今为止收集的与该检查有关的所有数据。 任何计划审核所使用的检查无法禁用。必须先从计划审核中删除检查，或删除计划审核自身。 第一次调用 `UpdateAccountAuditConfiguration` 时，必须提供此参数，且必须至少指定一个已启用的检查。
已启用	布尔值	如果已为此账户启用审核检查，则为 true。
configuration	映射	（可选）特定审计检查的自定义配置，例如 `CERT_AGE_THRESHOLD_IN_DAYS` 和 `CERT_EXPIRATION_THRESHOLD_IN_DAYS` ，可让您定义想何时收到有关证书期限和即将到期的提醒。

输出

无

错误

`InvalidRequestException`

请求的内容无效。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

DescribeAccountAuditConfiguration

获取此账户的 Device Defender 审核设置信息。设置包括审核通知的发送方式以及启用或禁用的审核检查类型。

摘要

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

`cli-input-json` 格式

```
{
}
```

输出

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

```
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

CLI 输出字段

名称	类型	描述
roleArn	字符串 最大长度：2048，最小长度：20	角色的 ARN，该角色授权 AWS IoT 在执行审核时访问设备、策略、证书及其它项相关信息。 第一次调用 `UpdateAccountAuditConfiguration` 时，必须提供此参数。
auditNotificationTargetConfigurations	映射	有关此账户的审核通知要发送到的目标的相关信息。
targetArn	字符串	审核通知要发送到的目标（SNS 主题）的 ARN。
roleArn	字符串 最大长度：2048，最小长度：20	可授权向目标发送通知的角色的 ARN。
已启用	布尔值	如果已启用向目标发送通知，则为 true。
auditCheckConfigurations	映射	针对此账户启用和禁用的审核检查。
已启用	布尔值	如果已为此账户启用审核检查，则为 true。

名称	类型	描述
configuration	映射	（可选）为某些审计检查提供特定配置，例如证书支持的最大期限或应触发提醒的到期前天数。

错误

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

DeleteAccountAuditConfiguration

还原此账户的默认 Device Defender 审核设置。您输入的所有配置数据将被删除，所有审计检查均重置为已禁用。

摘要

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

`cli-input-json` 格式

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

cli-input-json 字段

名称	类型	描述
deleteScheduledAudits	布尔值	如果为 True，则所有计划审核都将被删除。

输出

无

错误

`InvalidRequestException`

请求的内容无效。

`ResourceNotFoundException`

指定的资源不存在。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

计划审核

使用 `CreateScheduledAudit` 创建一个或多个计划审核。通过此命令，您可以指定在审核期间要执行的检查以及审核应该运行的频率。

使用 `ListScheduledAudits` 和 `DescribeScheduledAudit` 跟踪计划审核。

使用 `UpdateScheduledAudit` 更改现有的计划审核，或使用 `DeleteScheduledAudit` 将其删除。

CreateScheduledAudit

创建计划审核，使之按指定的时间间隔运行。

摘要

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

cli-input-json 格式

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

cli-input-json 字段

名称	类型	描述
frequency	字符串	计划审核的运行频率。可以是以下任一值：“DAILY”、“WEEKLY”、“BIWEEKLY”或“MONTHLY”。每个审核的实际开始时间取决于系统。 枚举：DAILY | WEEKLY | BIWEEKLY | MONTHLY

名称	类型	描述
dayOfMonth	字符串 模式：^([1-9]|[12][0-9]|3[01])$\|^LAST$	运行计划审核当月的具体日期。可以是“1”到“31”的任意值或“LAST”。如果 `frequency` 参数设置为“MONTHLY”，则此字段为必填字段。如果指定第 29-31 日，但当月没有那么多天，那么系统会在当月的最后一天运行审核。
dayOfWeek	字符串	运行计划审核当周的具体日期。可以是以下任一值：“SUN”、“MON”、“TUE”、“WED”、“THU”、“FRI”或“SAT”。如果 `frequency` 参数段设置为“WEEKLY”或“BIWEEKLY”，则此字段为必填字段。 枚举：SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	列表 成员：AuditCheckName	计划审核期间执行的检查。账户必须已启用检查。（使用 `DescribeAccountAuditConfiguration` 查看所有检查的列表，包括已启用的检查，或使用 `UpdateAccountAuditConfiguration` 选择启用的检查。）
tags	列表 成员：Tag java 类：java.util.List	可用于管理计划审核的元数据。
键	字符串	标签的键。

名称	类型	描述
值	字符串	标签的值。
scheduledAuditName	字符串 最大长度：128，最小长度：1 模式：[a-zA-Z0-9_-]+	要为计划审核指定的名称。（最多 128 个字符）

输出

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

CLI 输出字段

名称	类型	描述
scheduledAuditArn	字符串	计划审核的 ARN。

错误

`InvalidRequestException`

请求的内容无效。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

`LimitExceededException`

已超出限制。

ListScheduledAudits

列出所有计划审核。

摘要

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

cli-input-json 格式

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

cli-input-json 字段

名称	类型	描述
nextToken	字符串	用于获取下一组结果的令牌。
maxResults	整数 范围 - 最大值：250，最小值：1	一次性返回的最大结果数。默认值为 25。

输出

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

```
    }
  ],
  "nextToken": "string"
}
```

CLI 输出字段

名称	类型	描述
scheduledAudits	列表 成员：ScheduledAuditMetadata java 类：java.util.List	计划审核的列表。
scheduledAuditName	字符串 最大长度：128，最小长度：1 模式：[a-zA-Z0-9_-]+	计划审核的名称。
scheduledAuditArn	字符串	计划审核的 ARN。
frequency	字符串	计划审核的运行频率。 枚举：DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	字符串 模式：^([1-9]|[12][0-9]|3[01])$\|^LAST$	运行计划审核当月的具体日期（如果 `frequency` 为“MONTHLY”）。如果指定第 29-31 日，但当月没有那么多天，那么系统会在当月的最后一天运行审核。
dayOfWeek	字符串	运行计划审核当周的具体日期（如果 `frequency` 为“WEEKLY”或“BIWEEKLY”）。

名称	类型	描述
		枚举：SUN | MON | TUE | WED | THU | FRI | SAT
nextToken	字符串	用于检索下一组结果的令牌，没有更多结果时为 null。

错误

`InvalidRequestException`

请求的内容无效。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

DescribeScheduledAudit

获取有关计划审核的信息。

摘要

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

cli-input-json 格式

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

cli-input-json 字段

名称	类型	描述
scheduledAuditName	字符串 最大长度：128，最小长度：1 模式：[a-zA-Z0-9_-]+	要获取其信息的计划审核的名称。

输出

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

CLI 输出字段

名称	类型	描述
frequency	字符串	计划审核的运行频率。可以是以下任一值：“DAILY”、“WEEKLY”、“BIWEEKLY”或“MONTHLY”。每个审核的实际开始时间取决于系统。 枚举：DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	字符串 模式：^([1-9]|[12][0-9]|3[01])$\|^LAST$	运行计划审核当月的具体日期。可以是“1”到“31”的任意值或“LAST”。如果指定第 29-31 日，但当月没有那么多天，那

名称	类型	描述
		么系统会在当月的最后一天运行审核。
dayOfWeek	字符串	运行计划审核当周的具体日期。可以是以下任一值：“SUN”、“MON”、“TUE”、“WED”、“THU”、“FRI”或“SAT”。 枚举：SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	列表 成员：AuditCheckName	计划审核期间执行的检查。账户必须已启用检查。（使用 `DescribeAccountAuditConfiguration` 查看所有检查的列表，包括已启用的检查，或使用 `UpdateAccountAuditConfiguration` 选择启用的检查。）
scheduledAuditName	字符串 最大长度：128，最小长度：1 模式：[a-zA-Z0-9_-]+	计划审核的名称。
scheduledAuditArn	字符串	计划审核的 ARN。

错误

`InvalidRequestException`

请求的内容无效。

`ResourceNotFoundException`

指定的资源不存在。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

UpdateScheduledAudit

更新计划审核，包括执行的检查和审核执行的频率。

摘要

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

`cli-input-json` 格式

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

`cli-input-json` 字段

名称	类型	描述
frequency	字符串	计划审核的运行频率。可以是以下任一值：“DAILY”、“WEEKLY”、“BIWEEKLY

名称	类型	描述
		”或“MONTHLY”。每个审核的实际开始时间取决于系统。 枚举：DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	字符串 模式：^([1-9]|[12][0-9]|3[01])$\|^LAST$	运行计划审核当月的具体日期。可以是“1”到“31”的任意值或“LAST”。如果 `frequency` 参数设置为“MONTHLY”，则此字段为必填字段。如果指定第 29-31 日，但当月没有那么多天，那么系统会在当月的最后一天运行审核。
dayOfWeek	字符串	运行计划审核当周的具体日期。可以是以下任一值：“SUN”、“MON”、“TUE”、“WED”、“THU”、“FRI”或“SAT”。如果 `frequency` 参数段设置为“WEEKLY”或“BIWEEKLY”，则此字段为必填字段。 枚举：SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	列表 成员：AuditCheckName	计划审核期间执行的检查。账户必须已启用检查。（使用 `DescribeAccountAuditConfiguration` 查看所有检查的列表，包括已启用的检查，或使用 `UpdateAccountAuditConfiguration` 选择启用的检查。）

名称	类型	描述
scheduledAuditName	字符串 最大长度：128，最小长度：1 模式：[a-zA-Z0-9_-]+	计划审核的名称。（最多 128 个字符）

输出

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

CLI 输出字段

名称	类型	描述
scheduledAuditArn	字符串	计划审核的 ARN。

错误

`InvalidRequestException`

请求的内容无效。

`ResourceNotFoundException`

指定的资源不存在。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

DeleteScheduledAudit

删除计划审核。

摘要

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

cli-input-json 格式

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

cli-input-json 字段

名称	类型	描述
scheduledAuditName	字符串 最大长度：128，最小长度：1 模式：[a-zA-Z0-9_-]+	要删除的计划审核的名称。

输出

无

错误

`InvalidRequestException`

请求的内容无效。

`ResourceNotFoundException`

指定的资源不存在。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

运行按需审核

使用 StartOnDemandAuditTask 指定要执行的检查并立即开始运行审核。

StartOnDemandAuditTask

启动按需 Device Defender 审核。

摘要

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

cli-input-json 格式

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

cli-input-json 字段

名称	类型	描述
targetCheckNames	列表 成员：AuditCheckName	在审核过程中执行的检查。必须为账户启用指定的检查，否则会出现异常。（使用 DescribeAccountAuditConfiguration 查看所有检查的列表，包括已启用的检查，或使用 UpdateAccountAuditConfiguration 选择启用的检查。）

输出

```
{
```

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

CLI 输出字段

名称	类型	描述
taskId	字符串 最大长度：40，最小长度：1 模式：[a-zA-Z0-9-]+	启动的按需审核的 ID。

错误

`InvalidRequestException`

请求的内容无效。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

`LimitExceededException`

已超出限制。

管理审核实例

使用 `DescribeAuditTask` 获取关于特定审核实例的信息。如果审核已经运行，则结果会包括失败的检查、通过的检查、系统无法完成的检查，以及仍在运行中的检查（如果审核仍在进行中）。

使用 `ListAuditTasks` 查找在特定时间间隔内运行的审核。

使用 `CancelAuditTask` 停止正在进行的审核。

DescribeAuditTask

获取有关 Device Defender 审核的信息。

摘要

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

cli-input-json 格式

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

cli-input-json 字段

名称	类型	描述
taskId	字符串 最大长度：40，最小长度：1 模式：[a-zA-Z0-9-]+	要获取其信息的审核的 ID。

输出

```
{
  "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 输出字段

名称	类型	描述
taskStatus	字符串	审核的状态，可以是以下任一值：“IN_PROGRESS”、“COMPLETED”、“FAILED”或“CANCELED”。 枚举：IN_PROGRESS | COMPLETED | FAILED | CANCELED
taskType	字符串	审核的类型：“ON_DEMAND_AUDIT_TASK”或“SCHEDULED_AUDIT_TASK”。 枚举：ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStartTime	timestamp	审核的开始时间。
taskStatistics	TaskStatistics	有关审核的统计信息。
totalChecks	整数	此审核中检查的数量。
inProgressChecks	整数	正在进行的检查的数量。

名称	类型	描述
waitingForDataCollectionChecks	整数	等待数据收集的检查的数量。
compliantChecks	整数	发现合规资源的检查的数量。
nonCompliantChecks	整数	发现不合规资源的检查的数量。
failedChecks	整数	检查的数量。
canceledChecks	整数	因审核取消而未运行的检查的数量。
scheduledAuditName	字符串 最大长度：128，最小长度：1 模式：[a-zA-Z0-9_-]+	计划审核的名称（仅当审核是计划审核时）。
auditDetails	映射	有关在此审核过程中执行的每项检查的详细信息。
checkRunStatus	字符串	此检查的完成状态，可以是以下任一值：“IN_PROGRESS”、“WAITING_FOR_DATA_COLLECTION”、“CANCELED”、“COMPLETED_COMPLIANT”、“COMPLETED_NON_COMPLIANT”或“FAILED”。 枚举：IN_PROGRESS | WAITING_FOR_DATA_COLLECTION | CANCELED | COMPLETED_COMPLIANT | COMPLETED_NON_COMPLIANT | FAILED

名称	类型	描述
checkCompliant	布尔值	如果检查完成并找到所有合规资源，则为 True。
totalResourcesCount	长整数	已执行检查的资源的数量。
nonCompliantResourcesCount	长整数	检查发现不合规的资源的数量。
errorCode	字符串	在此审核过程中执行此检查时遇到的所有错误代码。可以是以下任一值：“INSUFFICIENT_PERMISSIONS”或“AUDIT_CHECK_DISABLED”。
message	字符串 最大长度：2048	与在此审核过程中执行此检查时遇到的所有错误相关的消息。

错误

`InvalidRequestException`

请求的内容无效。

`ResourceNotFoundException`

指定的资源不存在。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

ListAuditTasks

列出已在指定时间段内执行的 Device Defender 审核。

摘要

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

cli-input-json 格式

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

cli-input-json 字段

名称	类型	描述
startTime	timestamp	时间段的开始时间。审核信息只保留有限的时间（180 天）。在保留信息之前请求开始时间将引发 InvalidRequestException 。
endTime	timestamp	时间段的结束时间。
taskType	字符串	筛选条件，用于将输出限定到指定审核类型，可以是以下任一值：“ON_DEMAND_AUDIT_TASK”或“SCHEDULED__AUDIT_TASK”。

名称	类型	描述
		枚举：ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStatus	字符串	筛选条件，用于将输出限定到具有指定完成状态的审核，可以是以下任一值：“IN_PROGRESS”、“COMPLETED”、“FAILED”或“CANCELED”。 枚举：IN_PROGRESS | COMPLETED | FAILED | CANCELED
nextToken	字符串	用于获取下一组结果的令牌。
maxResults	整数 范围 - 最大值：250，最小值：1	一次性返回的最大结果数。默认值为 25。

输出

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

CLI 输出字段

名称	类型	描述
任务	列表 成员：AuditTaskMetadata java 类：java.util.List	在指定时间段内执行的审核。
taskId	字符串 最大长度：40，最小长度：1 模式：[a-zA-Z0-9-]+	此审核的 ID。
taskStatus	字符串	此审核的状态，可以是以下任一值：“IN_PROGRESS”、“COMPLETED”、“FAILED”或“CANCELED”。 枚举：IN_PROGRESS | COMPLETED | FAILED | CANCELED
taskType	字符串	此审核的类型，可以是以下任一值：“ON_DEMAND_AUDIT_TASK”或“SCHEDULED_AUDIT_TASK”。 枚举：ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
nextToken	字符串	用于检索下一组结果的令牌，没有更多结果时为 `null`。

错误

`InvalidRequestException`

请求的内容无效。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

CancelAuditTask

取消正在进行的审核。审核可能是计划审核，也可能是按需审核。如果审核未处于进行中，则将引发 `InvalidRequestException`。

摘要

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

`cli-input-json` 格式

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

`cli-input-json` 字段

名称	类型	描述
taskId	字符串 最大长度：40，最小长度：1 模式：[a-zA-Z0-9-]+	要取消的审核的 ID。只能取消状态为“IN_PROGRESS”的审核。

输出

无

错误

`ResourceNotFoundException`

指定的资源不存在。

`InvalidRequestException`

请求的内容无效。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

检查审核结果

使用 `ListAuditFindings` 查看审核结果。可以按照检查类型、特定资源或审核时间筛选结果。您可以使用此信息来解决发现的任何问题。

您可以定义缓解操作并将其应用于审核的结果。有关更多信息，请参阅 缓解操作。

ListAuditFindings

列出 Device Defender 审核的结果或在指定时间段内审核执行的结果。（结果保留 180 天。）

摘要

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

cli-input-json 格式

```
{
  "taskId": "string",
  "checkName": "string",
  "resourceIdentifier": {
    "deviceCertificateId": "string",
    "caCertificateId": "string",
    "cognitoIdentityPoolId": "string",
    "clientId": "string",
    "policyVersionIdentifier": {
      "policyName": "string",
      "policyVersionId": "string"
    },

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

cli-input-json 字段

名称	类型	描述
taskId	字符串 最大长度：40，最小长度：1 模式：[a-zA-Z0-9-]+	筛选条件，用于将结果限定到具有指定 ID 的审核。必须指定 taskId 或 startTime 和 endTime，但不能同时指定这两项。
checkName	字符串	筛选条件，用于将结果限定到指定审核检查结果。
resourceIdentifier	ResourceIdentifier	用于标识不合规资源的信息。
deviceCertificateId	字符串 最大长度：64，最小长度：64	附加到资源的证书的 ID。

名称	类型	描述
	模式：(0x)?[a-fA-F0-9]+	
caCertificateId	字符串 最大长度：64，最小长度：64 模式：(0x)?[a-fA-F0-9]+	授权证书所用的 CA 证书的 ID。
cognitoIdentityPoolId	字符串	Amazon Cognito 身份池的 ID。
clientId	字符串	客户端 ID。
policyVersionIdentifier	PolicyVersionIdentifier	与资源关联的策略的版本。
policyName	字符串 最大长度：128，最小长度：1 模式：[w+=,.@-]+	策略的名称。
policyVersionId	字符串 模式：[0-9]+	与资源关联的策略的版本 ID。
roleAliasArn	字符串	具有过于宽松操作的角色别名的 ARN。 最大长度：2048，最小长度：1
账户	字符串 最大长度：12，最小长度：12 模式：[0-9]+	资源所关联的账户。

名称	类型	描述
maxResults	整数 范围 - 最大值：250，最小值：1	一次性返回的最大结果数。默认值为 25。
nextToken	字符串	用于获取下一组结果的令牌。
startTime	timestamp	筛选条件，用于将结果限定到在指定时间之后发现的结果。必须指定 startTime 和 endTime 或 taskId，但不能同时指定这两项。
endTime	timestamp	筛选条件，用于将结果限定到在指定时间之前发现的结果。必须指定 startTime 和 endTime 或 taskId，但不能同时指定这两项。

输出

```
{
  "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 输出字段

名称	类型	描述
findings	列表 成员：AuditFinding	审核的结果。
taskId	字符串 最大长度：40，最小长度：1 模式：[a-zA-Z0-9-]+	生成此结果的审核的 ID。
checkName	字符串	生成此结果的审核检查。
taskStartTime	timestamp	审核的开始时间。
findingTime	timestamp	结果的发现时间。
severity	字符串	结果的严重性。 枚举：CRITICAL | HIGH | MEDIUM | LOW
nonCompliantResource	NonCompliantResource	经发现不符合审核检查规定的资源。
resourceType	字符串	不合规资源的类型。 枚举：DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	用于标识不合规资源的信息。
deviceCertificateId	字符串 最大长度：64，最小长度：64	附加到资源的证书的 ID。

名称	类型	描述
	模式：(0x)?[a-fA-F0-9]+	
caCertificateId	字符串 最大长度：64，最小长度：64 模式：(0x)?[a-fA-F0-9]+	授权证书所用的 CA 证书的 ID。
cognitoIdentityPoolId	字符串	Amazon Cognito 身份池的 ID。
clientId	字符串	客户端 ID。
policyVersionIdentifier	PolicyVersionIdentifier	与资源关联的策略的版本。
policyName	字符串 最大长度：128，最小长度：1 模式：[w+=,.@-]+	策略的名称。
policyVersionId	字符串 模式：[0-9]+	与资源关联的策略的版本 ID。
账户	字符串 最大长度：12，最小长度：12 模式：[0-9]+	资源所关联的账户。
additionalInfo	映射	有关不合规资源的其它信息。
relatedResources	列表 成员：RelatedResource	相关资源的列表。

名称	类型	描述
resourceType	字符串	资源的类型。 枚举：DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	用于标识资源的信息。
deviceCertificateId	字符串 最大长度：64，最小长度：64 模式：(0x)?[a-fA-F0-9]+	附加到资源的证书的 ID。
caCertificateId	字符串 最大长度：64，最小长度：64 模式：(0x)?[a-fA-F0-9]+	授权证书所用的 CA 证书的 ID。
cognitoIdentityPoolId	字符串	Amazon Cognito 身份池的 ID。
clientId	字符串	客户端 ID。
policyVersionIdentifier	PolicyVersionIdentifier	与资源关联的策略的版本。
iamRoleArn	字符串 最大长度：2048，最小长度：20	具有过于宽松操作的 IAM 角色的 ARN。
policyName	字符串 最大长度：128，最小长度：1 模式：[w+=,.@-]+	策略的名称。

名称	类型	描述
policyVersionId	字符串 模式：[0-9]+	与资源关联的策略的版本 ID。
roleAliasArn	字符串 最大长度：2048，最小长度：1	具有过于宽松操作的角色别名的 ARN。
账户	字符串 最大长度：12，最小长度：12 模式：[0-9]+	资源所关联的账户。
additionalInfo	映射	有关资源的其它信息。
reasonForNonCompliance	字符串	资源不合规的原因。
reasonForNonComplianceCode	字符串	用于表明资源不合规原因的代码。
nextToken	字符串	用于检索下一组结果的令牌，没有更多结果时为 null。

错误

`InvalidRequestException`

请求的内容无效。

`ThrottlingException`

速率超过限制。

`InternalFailureException`

出现意外错误。

审计查找结果隐藏

当您运行审计时，它会报告所有不合规资源的查找结果。这意味着您的审计报告包含您正在努力缓解问题的资源的查找结果以及已知不合规资源（如测试设备或损坏设备）的查找结果。审计将继续报告在连续审计运行中仍然不合规的资源的查找结果，这可能会向您的报告中添加不需要的信息。使用审计查找结果，您可以在定义的时间段内隐藏或筛选出查找结果，直到资源的问题被解决，若是针对与测试或损坏设备关联的资源，则可无限期地隐藏结果。

> **Note**
>
> 缓解操作不适用于被隐藏的审计结果。有关缓解操作的更多信息，请参阅 缓解操作。

有关审计检查结果隐藏配额的信息，请参阅 AWS IoT Device Defender 端点与配额。

审计查找结果隐藏的工作原理

当您为不合规的资源创建审计查找结果隐藏时，您的审计报告和通知的行为会有所不同。

您的审计报告将包含一个新部分，其中列出与报告关联的所有隐藏的查找结果。当我们评估审计检查是否合规时，不会考虑隐藏查找结果。当您在命令行界面 (CLI) 中使用 describe-audit-task 命令时，每个审计检查也会返回隐藏的资源计数。

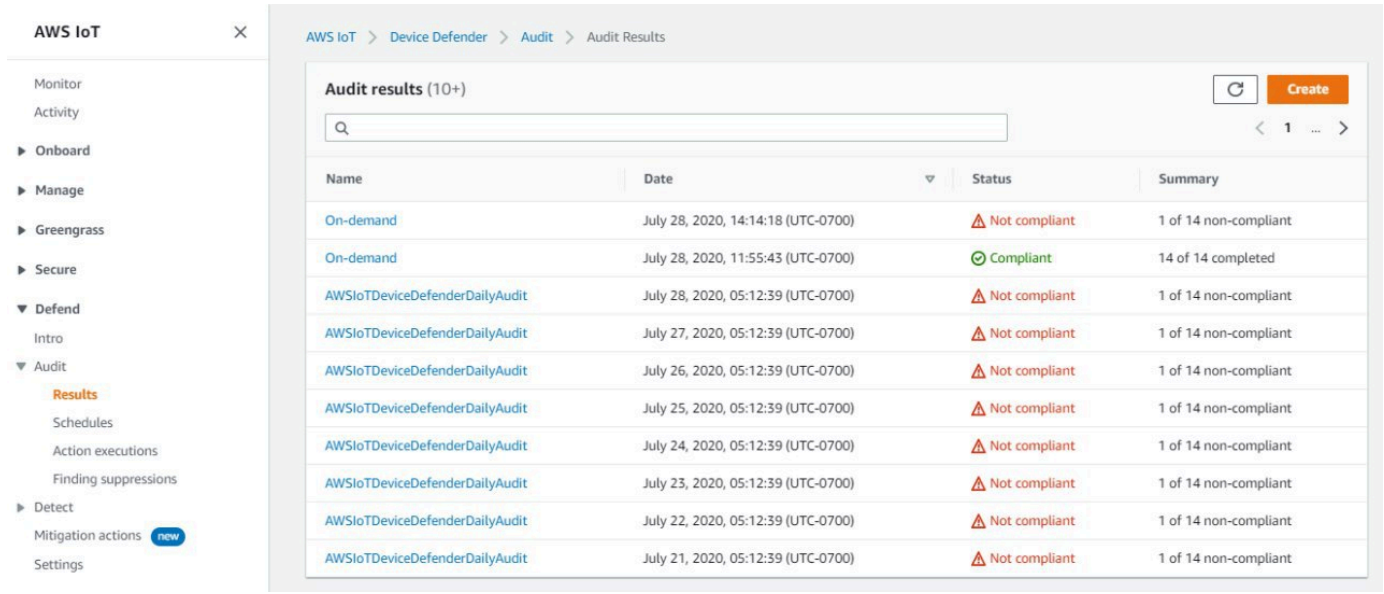
对于审计通知，当我们评估审计检查是否合规时，不会考虑隐藏的查找结果。AWS IoT Device Defender 发布到 Amazon CloudWatch 和 Amazon Simple Notification Service (Amazon SNS) 的每个审计检查通知中也包含隐藏资源计数。

如何在控制台中使用审计查找结果隐藏

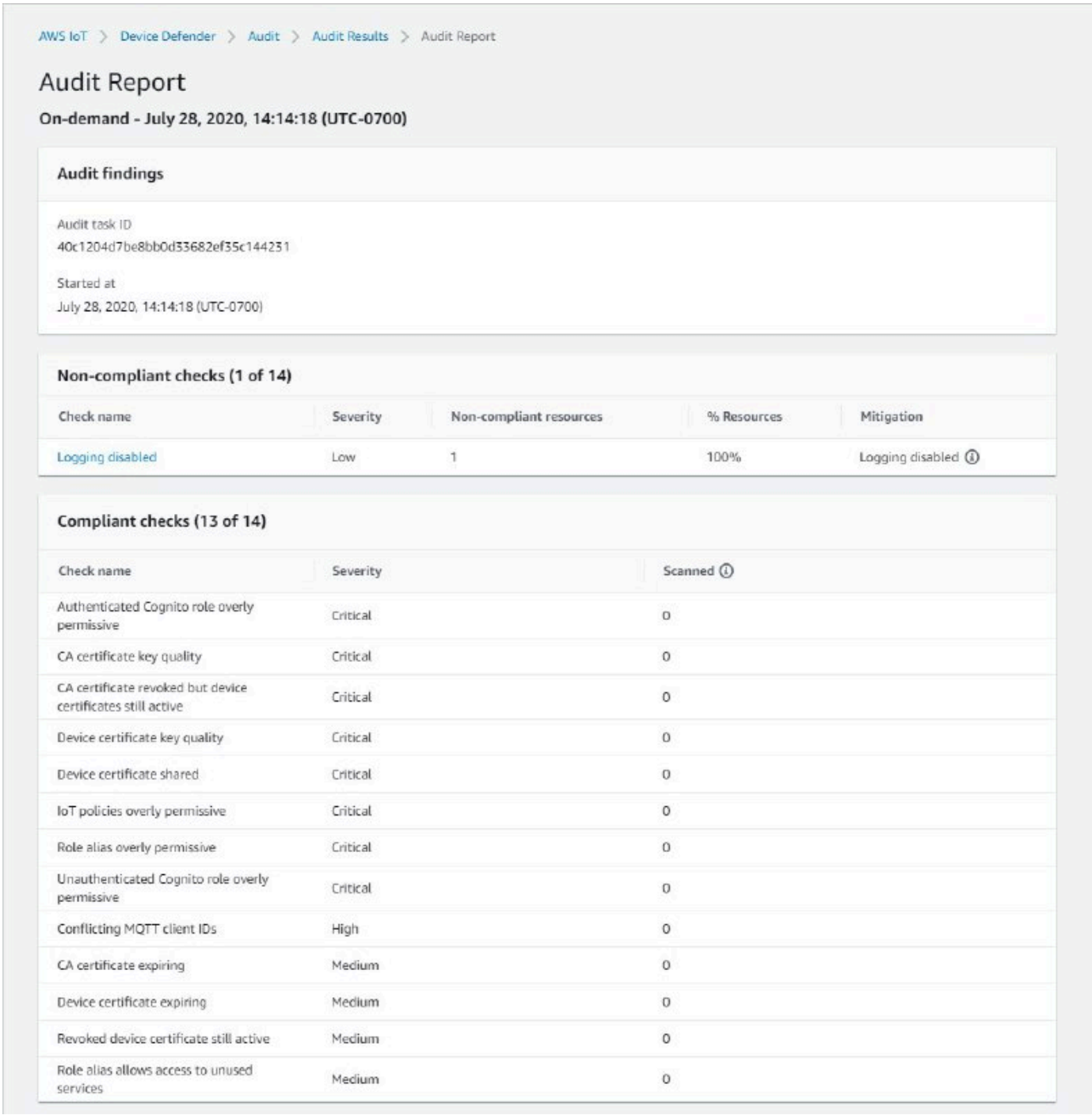
要隐藏审计报告中的查找结果

以下流程介绍了如何在 AWS IoT 控制台中创建审计查找结果隐藏。

1. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Audit（审计）、Results（结果）。
2. 选择您要查看的审计报告。



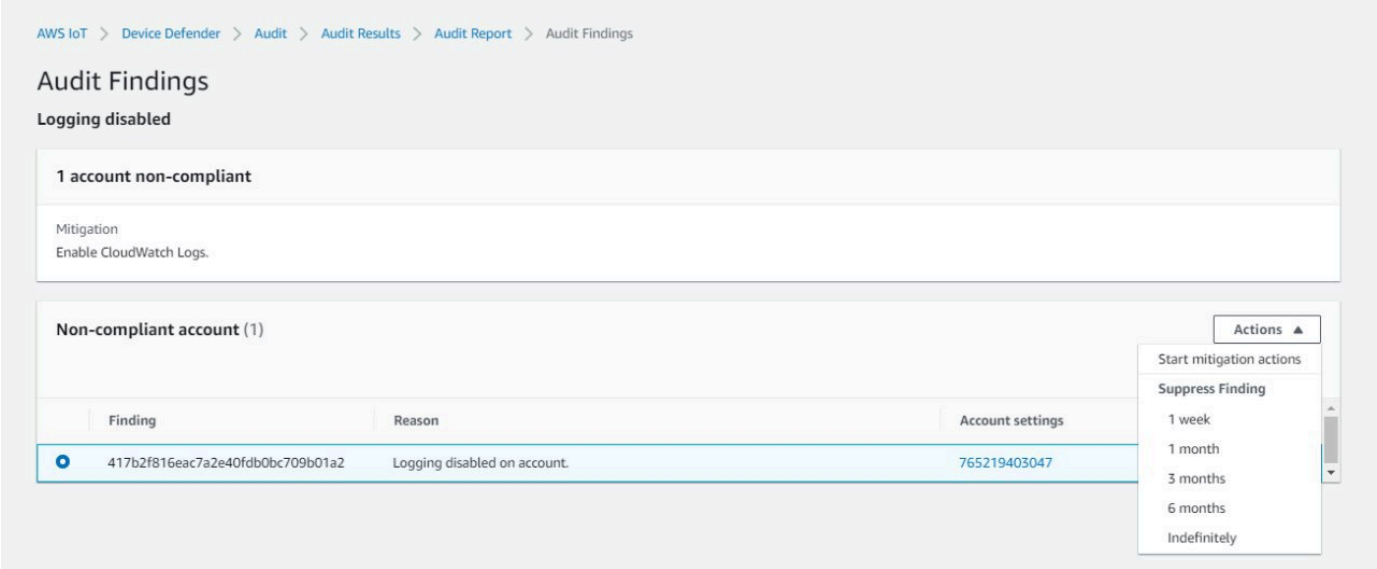
3. 在 Non-compliant checks（不合规检查）部分，在 Check name（检查名称）项下，选择您感兴趣的审计检查。



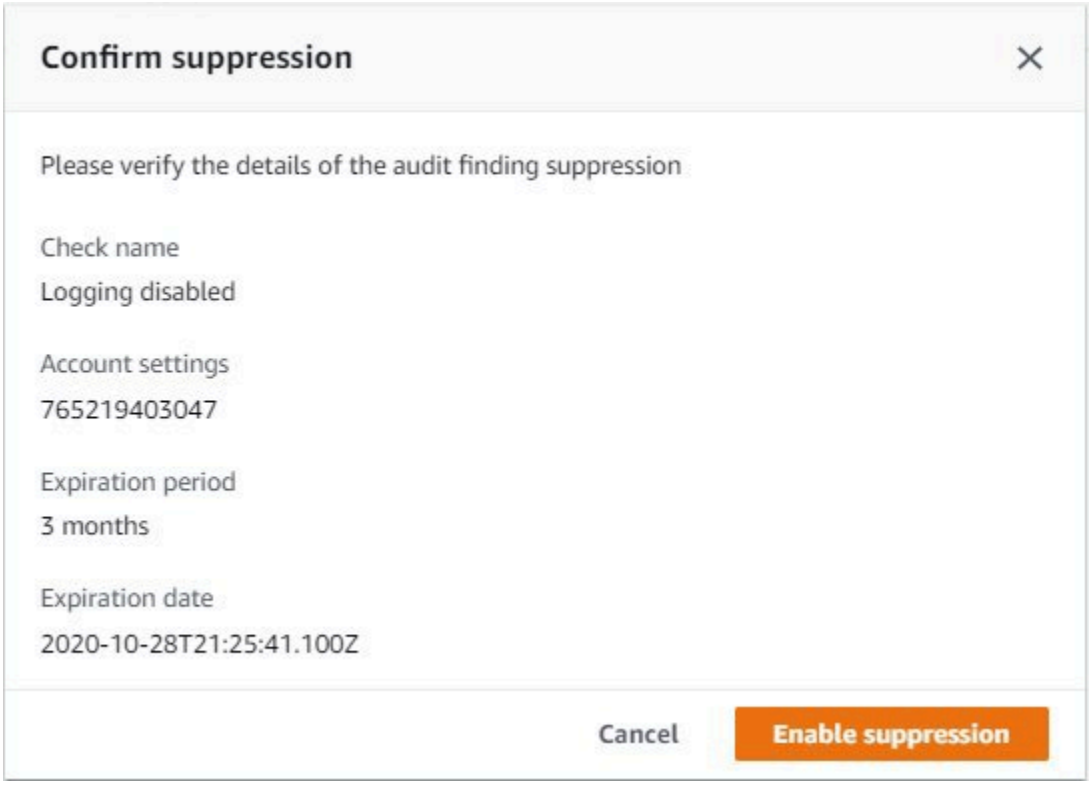
4. 在审计检查详细信息屏幕上，如果存在您不想看到的查找结果，请选择查找结果旁边的选项按钮。然后，选择 Actions（操作），接着选择您希望审计查找结果隐藏持续的时长。

> **Note**
>
> 您可以在控制台中选择 1 week（1 周）、1 month（1 个月）、3 months（3 个月）、6 months（6 个月）或 Indefinitely（无限期），作为审计查找结果隐藏的到期日期。如果要

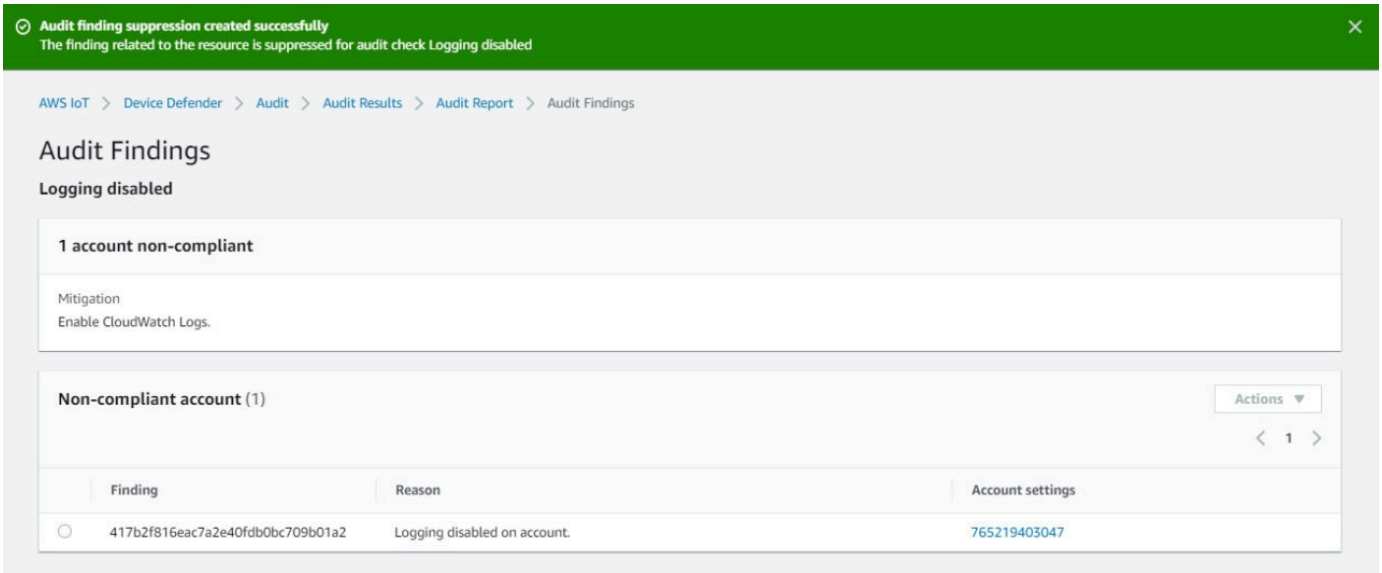
设置特定的到期日期，则只能在 CLI 或 API 中执行此操作。无论到期日期为何，您都可以随时取消审计查找结果隐藏。



5. 确认隐藏详细信息，然后选择 Enable suppresion（启用隐藏）。

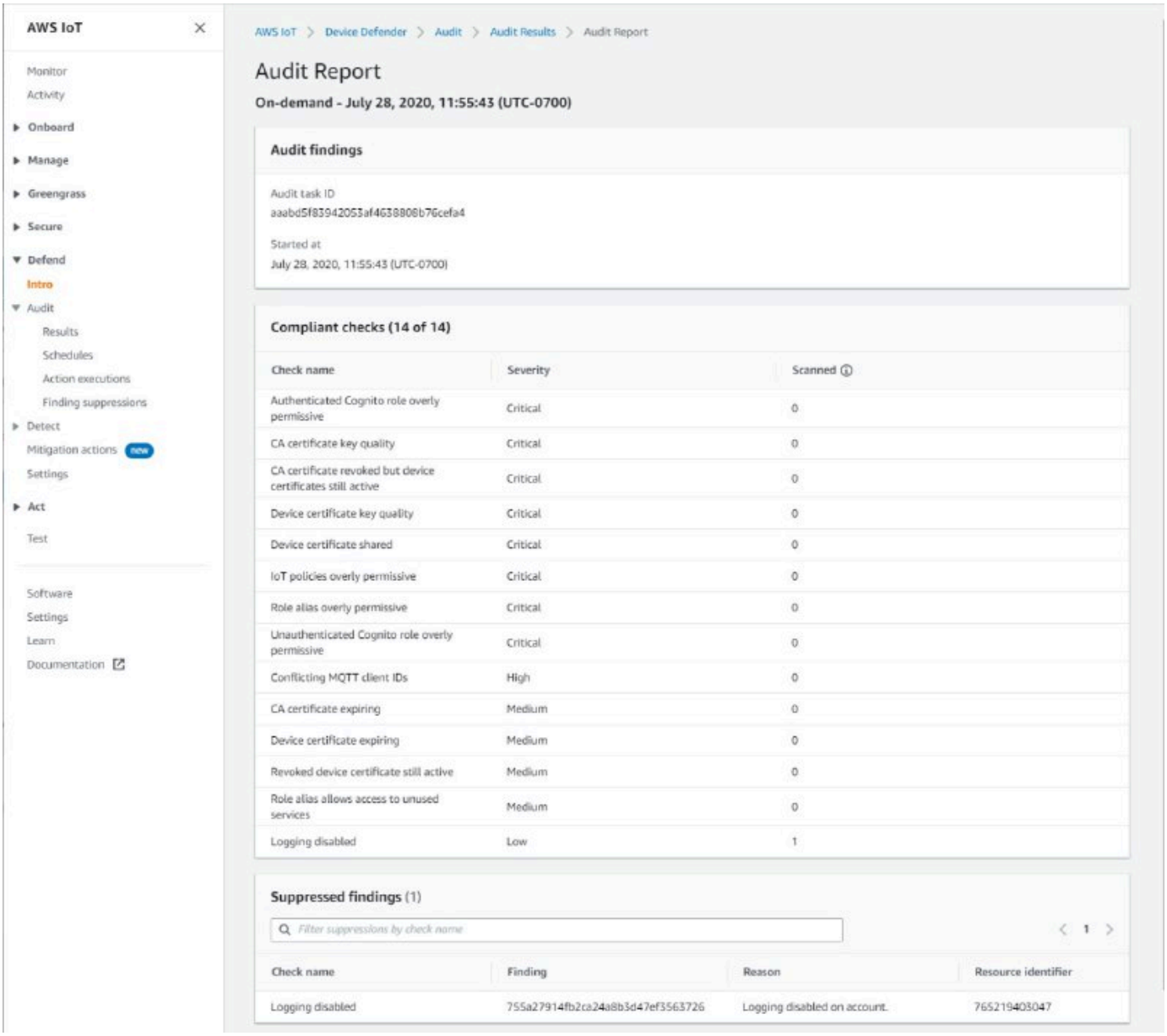


6. 创建审计查找结果隐藏后，将显示一个提示，确认审计查找结果隐藏已创建。



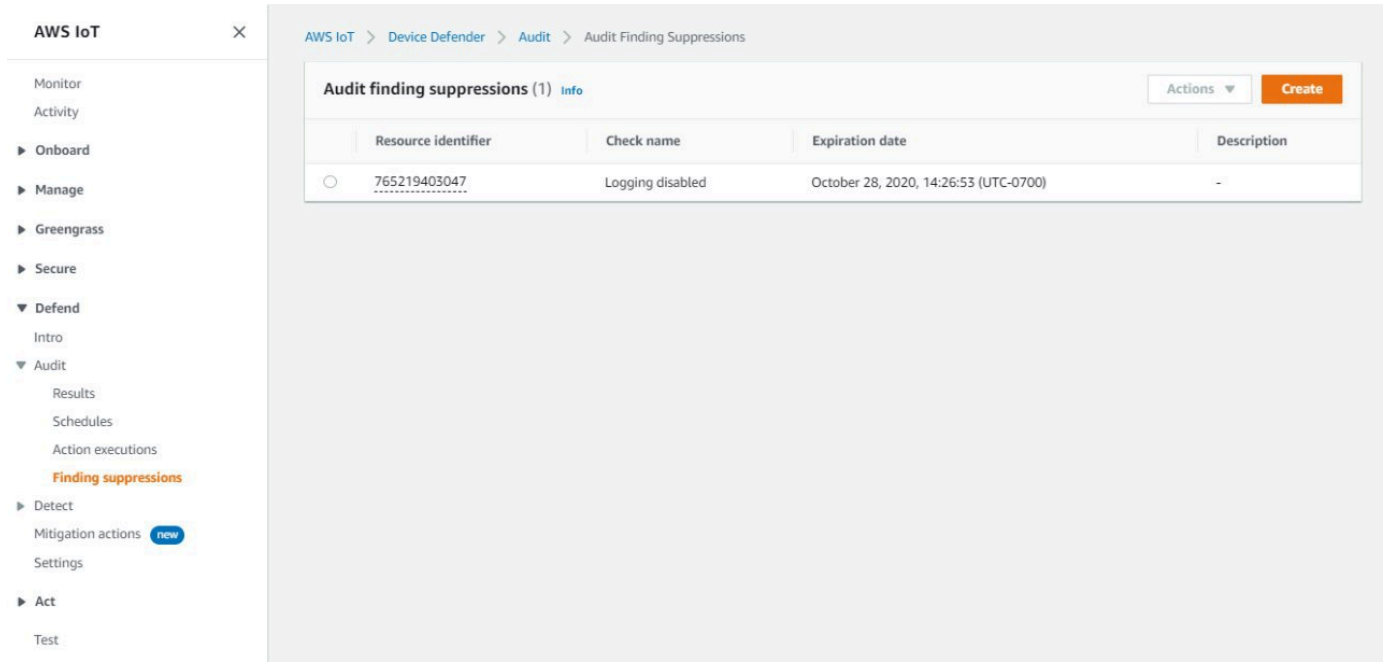
在审计报告中查看隐藏的结果

1. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Audit（审计）、Results（结果）。
2. 选择您要查看的审计报告。
3. 在 Suppressed findings（隐藏查找结果）部分，查看已针对您选择的审计报告隐藏了哪些审计结果。



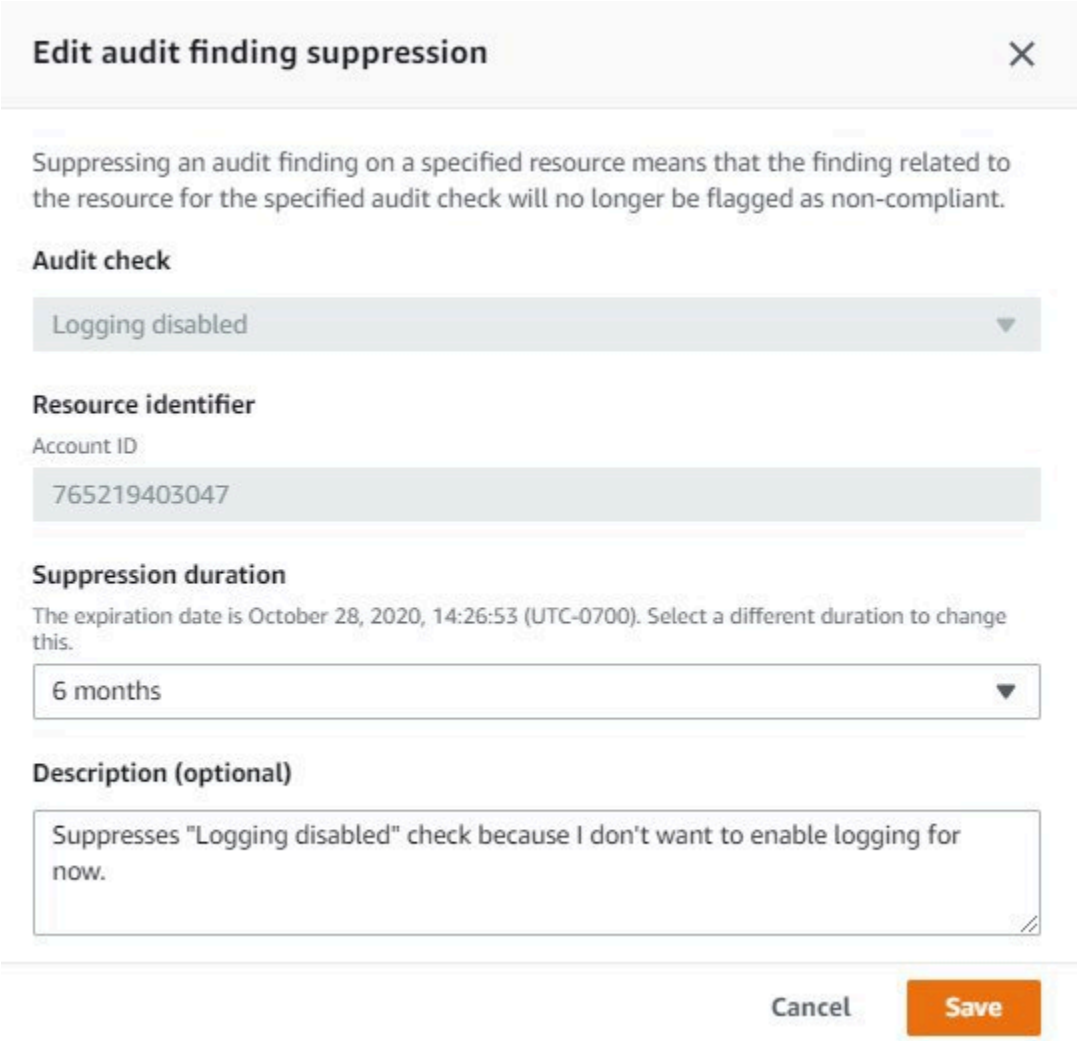
列出审计查找结果隐藏

- 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Audit（审计）、Finding suppressions（查找结果隐藏）。



要编辑审计查找结果隐藏

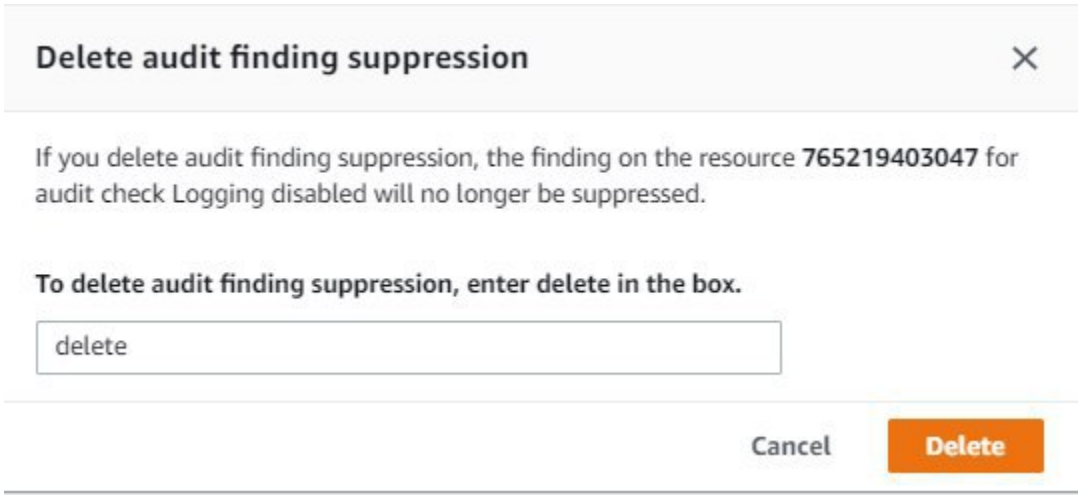
1. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Audit（审计）、Finding suppressions（查找结果隐藏）。
2. 选择要编辑的审计查找结果隐藏旁边的选项按钮。下一步，选择 Actions（操作）、Edit（编辑）。
3. 在 Edit audit finding suppression（编辑查找结果隐藏）窗口中，您可以更改 Suppression duration（隐藏时长）或者 Description (optional)（描述（可选））。



4. 执行您的更改后，选择 Save（保存）。Finding suppressions（查找结果隐藏）窗口将打开。

要删除审计查找结果隐藏

1. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Audit（审计）、Finding suppressions（查找结果隐藏）。
2. 选择要删除的审计查找结果隐藏旁边的选项按钮，然后选择 Actions（操作）、Delete（删除）。
3. 在 Delete audit finding suppression（删除审计查找结果隐藏）窗口中，在文本框中输入 `delete` 以确认删除，然后选择 Delete（删除）。Finding suppressions（查找结果隐藏）窗口将打开。



如何在 CLI 中使用审计查找结果隐藏

您可以使用以下 CLI 命令来创建和管理审计查找结果隐藏。

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

您输入的 `resource-identifier` 取决于您需要隐藏查找结果的 `check-name`。下表详细说明了哪些检查需要哪个 `resource-identifier` 来用于创建和编辑隐藏。

> **Note**
> 隐藏命令不意味着关闭审计。审计仍会在您的 AWS IoT 设备上运行。隐藏仅适用于审计结果。

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

要创建和应用审计查找结果隐藏

以下流程介绍如何在 AWS CLI 中创建审计查找结果隐藏。

- 使用 `create-audit-suppression` 命令创建审计查找结果隐藏。以下示例为根据 Logging disabled（禁用的日志记录）这项检查为 AWS 账户 *123456789012* 创建了审计查找结果隐藏。

```
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."
```

此命令无任何输出。

审计查找结果隐藏 API

以下 API 可用于创建和管理审计查找结果隐藏。

- CreateAuditSuppression
- DescribeAuditSuppression
- UpdateAuditSuppression
- DeleteAuditSuppression
- ListAuditSuppressions

要对特定审计查找结果进行筛选，您可以使用 ListAuditFindings API。

Detect

借助 AWS IoT Device Defender Detect，您可以监控设备的行为，以识别可能表明设备遭到危害的异常行为。使用云端指标（来自 AWS IoT）和设备端指标（来自您在设备上安装的代理）的组合，您可以检测到：

- 连接模式的变化。
- 与未经授权或无法识别的终端节点通信的设备。
- 入站和出站设备流量模式的变化。

您可以创建安全配置文件（其中包含设备预期行为的定义），并将它们分配到实例集中的一组设备或所有设备。AWS IoT Device Defender Detect 使用这些安全配置文件来检测异常，并通过 Amazon CloudWatch 指标和 Amazon Simple Notification Service 通知发送提醒。

AWS IoT Device Defender Detect 能够检测相连设备中频繁出现的安全问题：

- 从设备到已知恶意 IP 地址，或表示潜在恶意命令和控制渠道的未经授权终端节点的流量。
- 表明设备正在参与 DDoS 攻击的恶意流量，例如出站流量高峰。
- 其远程管理接口和端口支持远程访问的设备。
- 发送到您账户的消息速率激增（例如，来自可能导致过量按每条消息收费的流氓设备）。

使用案例：

评估攻击面

可以使用 AWS IoT Device Defender Detect 评估设备的攻击面。例如，可以识别服务端口通常成为攻击活动目标的设备（在端口 23/2323 上运行的 telnet 服务，在端口 22 上运行的 SSH 服务，在端口 80/443/8080/8081 上运行的 HTTP/S 服务）。虽然在设备上使用这些服务端口可能有合法的原因，但是它们也通常是敌人攻击面的一部分，并且具有相关风险。在 AWS IoT Device Defender Detect 向您发出攻击面告警后，您可以将攻击面降至最低（通过消除未使用的网络服务），或运行其它评估来识别安全漏洞（例如，使用常见、默认或弱密码配置的 telnet）。

检测设备行为异常以及可能的安全根本原因

您可以使用 AWS IoT Device Defender Detect 提醒您可能表明安全违规的意外设备行为指标（开放端口数量、连接数量、意外开放端口、与意外 IP 地址的连接）。例如，TCP 连接数量高于预期，

可能表明设备正被用于 DDoS 攻击。侦听的端口不是预计的端口，可能表明设备上安装了用以进行远程控制的后门。您可以使用 AWS IoT Device Defender Detect 探测设备实例集的运行状况，并验证安全假设（例如，没有设备在侦听端口 23 或 2323）。

您可以启用基于机器学习 (ML) 的威胁检测，以自动识别潜在威胁。

检测配置错误的设备

从设备发送到账户的信息数量或大小激增，可能表明设备配置错误。此类设备可能会增加每条消息的费用。同样，如果设备多次授权失败，则可能要求重新配置策略。

监控未注册设备的行为

AWS IoT Device Defender Detect 使得识别未在 AWS IoT 注册表中注册的设备的异常行为成为可能。您可以定义特定于以下目标类型之一的安全配置文件：

- 所有设备
- 所有已注册的设备（AWS IoT 注册表中的事物）
- 所有未注册的设备
- 事物组中的设备

安全配置文件为账户中的设备定义一组预期行为，并指定在检测到异常时要执行的操作。安全配置文件应附加到最具体的目标，以便您精细控制要针对该配置文件评估哪些设备。

未注册的设备在整个设备生命周期内必须提供一致的 MQTT 客户端标识符或事物名称（对于报告设备指标的设备），以便所有违规和指标都归属到相同设备。

> **Important**
>
> 如果事物名称包含控制字符，或事物名称的长度超过 128 字节的 UTF-8 编码字符，则将拒绝设备报告的消息。

安全使用案例

本部分介绍威胁您设备机群的不同类型的攻击，以及可用于监控这些攻击的建议指标。我们建议使用指标异常作为调查安全问题的起始操作，但您不应仅仅根据指标异常来确定任何安全威胁。

要调查异常告警，请将告警详细信息与其它上下文信息（如设备属性、设备指标历史趋势、安全配置文件指标历史趋势、自定义指标和日志）相关联，以确定是否存在安全威胁。

云端使用案例

Device Defender 可以在 AWS IoT 云端监控以下用例。

知识产权盗窃：

知识产权盗窃涉及盗窃个人或公司的知识产权，包括商业秘密、硬件或软件。它经常发生在设备的制造阶段。知识产权盗窃可能以盗版、设备盗窃或设备证书盗窃的形式出现。由于存在允许意外访问 IoT 资源的策略，因此可能会发生基于云的知识产权盗窃。您应检视您的 IoT 策略并启用审计过于宽容的检查以确定过于宽容的策略。

相关指标：

指标	理由
源 IP	如果设备被盗，则其源 IP 地址将超出正常供应链中流通的设备的正常预期 IP 地址范围。
已收到消息的数量	由于攻击者可能会在基于云的 IP 窃取中使用设备，因此与从 AWS IoT 云发送至设备的消息计数或消息大小有关的指标可能会激增，这表明可能存在安全问题。
消息大小	

基于 MQTT 的数据泄漏：

当恶意行为者从 IoT 部署或设备执行未经授权的数据传输时，就会发生数据泄露。攻击者通过 MQTT 对云端数据源发起此类攻击。

相关指标：

指标	理由
源 IP	如果设备被盗，则其源 IP 地址将超出标准供应链中流通设备的正常预期 IP 地址范围。
已收到消息的数量	由于攻击者可能会在基于 MQTT 的数据泄漏中使用设备，因此与从 AWS IoT 云发送至设备

指标	理由
消息大小	消息计数或消息大小有关的指标可能会激增，这表明可能存在安全问题。

模拟：

模拟攻击是指攻击者装作已知或受信任的实体，试图访问 AWS IoT 云端服务、应用程序、数据或对 IoT 设备发布命令和加以控制。

相关指标：

指标	理由
授权失败	当攻击者使用被盗的身份装作受信任实体时，与连接相关的指标往往会激增，因为凭证可能不再有效，或者可能已经被信任设备使用。授权失败、连接尝试或断开连接中的异常行为通常指向潜在的模拟攻击情况。
连接尝试次数	
断开连接	

云基础设施滥用：

AWS IoT 云服务的滥用在发布或订阅消息量较大的主题或大型消息的主题时会发生。过度宽容的策略或设备漏洞攻击针对命令和控制，也可能导致云基础设施滥用。这次攻击的主要目标之一是增加您的AWS账单。您应检视您的 IoT 策略并启用审计过于宽容的检查以确定过于宽容的策略。

相关指标：

指标	理由
已收到消息的数量	这次攻击的目的是增加您的AWS账单，监控消息计数、收到的消息和消息大小等活动的指标将会激增。
已发送消息的数量	
消息大小	
源 IP	可能会出现可疑的源 IP 列表，攻击者将从中生成其消息收发量。

设备端使用案例

Device Defender 可以在您的设备端监控以下使用案例。

拒绝服务攻击：

拒绝服务 (DoS) 攻击旨在关闭设备或网络，使目标用户无法访问该设备或网络。DoS 攻击通过使目标流量泛滥，或发送请求以使系统启动速度减慢或导致系统失败来阻止访问。您的 IoT 设备可用于 DoS 攻击。

相关指标：

指标	理由
传出的数据包	DoS 攻击通常涉及来自给定设备较高的出站通信速率，而且根据 DoS 攻击的类型，输出数据包数和输出字节数可能会单独增加或同时增加。
传出的字节数	
目的地 IP	如果您定义了设备应与之通信的 IP 地址 / CIDR 范围，则目标 IP 中的异常可能表示设备出现未经授权的 IP 通信。
侦听 TCP 端口	DoS 攻击通常需要更大的命令和控制基础设施，在此基础设施中，安装在设备上的恶意软件将接收有关攻击者和攻击时间的命令和信息。因此，为了接收此类信息，恶意软件通常会侦听一般不被设备使用的端口。
侦听 TCP 端口计数	
侦听 UDP 端口	
侦听 UDP 端口计数	

横向威胁升级：

横向威胁升级通常从攻击者获得对网络中某个点（例如连网的设备）的访问权限开始。随后，攻击者会试图通过被盗凭证或漏洞利用等方法提高其权限级别或对其它设备的访问权限。

相关指标：

指标	理由
传出的数据包	在典型情况下，攻击者必须在局域网上运行扫描，以执行侦测并识别可用设备，从而缩小攻击目标的选择范围。这种扫描可能会导致字节激增和数据包输出计数。
传出的字节数	
目的地 IP	如果设备将与一组已知的 IP 地址或 CIDR 通信，您可以确定它是否尝试与异常 IP 地址进行通信，在横向威胁升级的使用案例中，该 IP 地址通常是本地网络上的私有 IP 地址。
授权失败	当攻击者试图提高他们在 IoT 网络中的权限级别时，他们可能会使用被撤销或已过期的被盗凭证，从而导致更多的授权失败。

数据泄露或监控：

当恶意软件或恶意参与者从设备或网络端点进行未经授权的数据传输时，就会发生数据泄露。对于攻击者来说，数据泄露通常有两种目的：获取数据或知识产权，或者对网络进行侦测。监控意味着恶意代码被用于监控用户活动，其目的是窃取凭证和收集信息。以下指标可作为调查任一类型攻击的起点。

相关指标：

指标	理由
传出的数据包	当数据泄露或监控攻击发生时，攻击者通常会对从设备发送的数据进行镜像处理，而不是简单地重新导向数据，当防御程序看不到预期数据时，这些操作将被防御程序识别。此类镜像数据会显著增加从设备发送的数据总量，从而导致输出数据包和字节计数激增。
传出的字节数	
目的地 IP	当攻击者使用设备进行数据泄露或监控攻击时，必须将数据发送到攻击者控制的异常 IP 地址。监控目标 IP 有助于识别此类攻击。

加密数字货币挖掘

攻击者利用设备的处理能力挖掘加密数字货币。加密挖掘是一个计算密集型过程，通常需要与其它采矿同行和池进行网络通信。

相关指标：

指标	理由
目的地 IP	加密挖掘过程中通常需要有网络通信。拥有一个严格控制的、设备应与之通信的 IP 地址列表可以帮助识别设备上的意外通信，如加密数字货币挖掘。
CPU 使用率自定义指标	加密数字货币挖掘需要依靠密集的计算，这会导致设备 CPU 出现高利用率。如果您选择收集和监控此指标，则高于正常的 CPU 使用率可能是加密挖掘活动的迹象。

命令和控制、恶意软件和勒索软件

恶意软件或勒索软件会限制您对设备的控制，并限制您的设备功能。在出现勒索软件攻击的情况下，数据访问将会因勒索软件使用的加密手段而丢失。

相关指标：

指标	理由
目的地 IP	网络或远程攻击占 IoT 设备攻击的很大一部分。严格控制的、设备应与之通信的 IP 地址列表有助于识别恶意软件或勒索软件攻击导致的异常目标 IP。
侦听 TCP 端口	多种恶意软件攻击涉及启动命令和控制服务器，该服务器将发送命令以在设备上执行。此类服务器对恶意软件或勒索软件操作至关重要，可通过严密监控打开的 TCP/UDP 端口和端口计数来加以识别。
侦听 TCP 端口计数	
侦听 UDP 端口	
侦听 UDP 端口计数	

概念

指标

AWS IoT Device Defender Detect 使用指标来检测设备的异常行为。AWS IoT Device DefenderDetect 将指标的报告值与您提供的预期值进行比较。这些指标可以从两个来源获取：云端指标和设备端指标。ML Detect 支持 6 个云端指标和 7 个设备端指标。有关 ML Detect 支持指标的列表，请参阅 受支持的指标。

使用云端指标（例如，授权失败次数、设备通过 AWS IoT 发送或接收的消息数量或大小）检测 AWS IoT 网络中的异常行为。

AWS IoT Device Defender Detect 也可以收集、聚合和监控 AWS IoT 设备生成的指标数据（例如，设备侦听的端口、发送的字节或数据包数量，或设备的 TCP 连接）。

可以只将 AWS IoT Device Defender Detect 与云端指标相结合使用。要使用设备端指标，必须首先在连接 AWS IoT 的设备或设备网关上部署 AWS IoT SDK，以收集指标并发送到 AWS IoT。请参阅从设备发送指标。

安全配置文件

安全配置文件为账户中的一组设备（静态事物组）或所有设备定义异常行为，并指定在检测到异常时要执行的操作。您可以使用 AWS IoT 控制台或 API 命令创建安全配置文件，并将其与一组设备关联。AWS IoT Device DefenderDetect 开始记录安全相关数据，并使用安全配置文件中定义的行为检测设备行为中的异常情况。

行为

行为告诉 AWS IoT Device Defender Detect 如何识别设备是否发生异常。任何设备操作与行为不匹配时，均会触发提示。规则检测行为由指标和绝对值或统计阈值以及运算符组成（例如，小于等于、大于等于），用于描述预期的设备行为。ML Detect 行为由指标和 ML Detect 配置组成，这些行为可以设置 ML 模型来了解设备的正常行为。

ML 模型

ML 模型是一种机器学习模型，用于监控客户配置的每个行为。该模型根据目标设备组的指标数据模式进行训练，并为基于指标的行为生成三个异常置信阈值（高、中和低）。它根据设备级别的摄取指标数据推断异常。在 ML Detect 上下文中，将创建一个 ML 模型来评估一个基于指标的行为。有关更多信息，请参阅ML Detect。

置信级别

ML Detect 支持三个置信级别：`High`、`Medium` 和 `Low`。`High` 置信度意味着异常行为评估的灵敏度较低，而且告警数量往往较少。`Medium` 置信度意味着中等灵敏度，`Low` 置信度意味着高灵敏度和通常更高的告警数量。

维度

您可以定义维度以调整行为的作用域。例如，您可以定义一个主题筛选条件维度，该维度将行为应用于与模式匹配的 MQTT 主题。有关定义要在安全配置文件中使用的维度的信息，请参阅 CreateDimension。

告警

检测到异常时，系统会通过 CloudWatch 指标（请参阅《AWS IoT Core 开发人员指南》中的使用 Amazon CloudWatch 监控 AWS IoT 警报和指标）或 SNS 通知发送警报通知。AWS IoT 控制台中也会显示告警通知，以及告警的相关信息和设备告警历史记录。当监控的设备不再表现出异常行为，或者已经触发告警但在较长时间内停止报告时，系统也会发送告警。

告警验证状态

创建告警后，您可以验证告警为真、良、误报或未知。您还可以为告警验证状态添加描述。您可以通过四种验证状态之一查看、组织和筛选 AWS IoT Device Defender 告警。您可以使用告警验证状态和相关说明来通知团队成员。这有助于您的团队采取后续行动，例如，对真告警执行缓解措施、跳过良性告警或继续对未知告警进行调查。所有告警的原定设置验证状态为 “未知”。

告警隐藏

通过将行为通知设置为 on 或者 `suppressed` 来管理 Detect 告警 SNS 通知。隐藏告警不会阻止 Detect 执行设备行为评估；Detect 会继续将异常行为标记为违规告警。但是，隐藏的告警不会转发 SNS 通知。它们只能通过 AWS IoT 控制台或 API 访问。

行为

安全配置文件包含一组行为。每个行为都包含一个指标，用于指定账户中的一组设备或所有设备的正常行为。行为分为两类：Rule Detect 行为和 ML Detect 行为。使用 Rules Detect 行为，您可以定义设备的行为方式，而 ML Detect 使用基于历史设备数据构建的 ML 模型来评估设备应有的行为方式。

安全配置文件可以是以下两种阈值类型之一：ML 或者 基于规则。ML 安全配置文件会借助过去的数据自动检测机群中的设备级操作和安全异常。基于规则的安全配置文件要求您手动设置静态规则以监控您的设备行为。

下面描述了 behavior 定义中使用的一些字段：

Rules Detect 和 ML Detect 的共同点

name

行为的名称。

metric

所用指标的名称（即行为评估的对象）。

consecutiveDatapointsToAlarm

如果设备在指定数量的连续数据点违反了行为，则会发出警报。如果未指定，默认值为 1。

consecutiveDatapointsToClear

如果发出警报后，违规设备不再违反指定数量的连续数据点的行为，则警报将被解除。如果未指定，默认值为 1。

threshold type

安全性配置文件可以是两种阈值类型之一：ML 或基于规则。ML 安全配置文件会借助过去的数据自动检测机群中的设备级操作和安全异常。基于规则的安全配置文件要求您手动设置静态规则以监控您的设备行为。

alarm suppressions

您可以通过将行为通知设置为 on 或 suppressed 来管理 Detect 警报 Amazon SNS 通知。隐藏告警不会阻止 Detect 执行设备行为评估；Detect 会继续将异常行为标记为违规告警。然而，不会针对 Amazon SNS 通知转发隐藏的警报。它们只能通过 AWS IoT 控制台或 API 进行访问。

Rule Detect

dimension

您可以定义维度以调整行为的作用域。例如，您可以定义一个主题筛选条件维度，该维度将行为应用于与模式匹配的 MQTT 主题。要定义要在安全配置文件中使用的维度，请参阅 CreateDimension。仅适用于 Rules Detect。

criteria

确定设备与 metric 相关的行为是否正常的标准。

> **Note**
>
> 在 AWS IoT 控制台中，您可以选择提醒我，以便在 AWS IoT Device Defender 检测到设备行为异常时通过 Amazon SNS 收到通知。

`comparisonOperator`

将评估的事物 (`metric`) 与标准（`value` 或 `statisticalThreshold`）关联在一起的运算符。

可能的值为：“less-than”、“less-than-equals”、“greater-than”、“greater-than-equals”、“in-cidr-set”、“not-in-cidr-set”、“in-port-set”和“not-in-port-set”。并非所有运算符对于每个指标都是有效的。针对 CIDR 集和端口的运算符仅用于与此类实体有关的指标。

`value`

与 `metric` 进行比较的值。根据不同的指标类型，应当包含 `count`（一个值）、`ports`（CIDR 列表）或 `cidrs`（端口列表）。

`statisticalThreshold`

作为行为违规判断依据的统计阈值。此字段包含一个 `statistic` 字段，后者具有以下可能的值：“p0”、“p0.1”、“p0.01”、“p1”、“p10”、“p50”、“p90”、“p99”、“p99.9”、“p99.99”或“p100”。

此 `statistic` 表示一个百分位数。它解析为用于确定符合行为的值。在指定持续时间 (`durationSeconds`) 内从与此安全配置文件关联的所有报告设备收集指标一次或多次，并根据该数据计算百分位数。之后，收集设备的测量值并在相同的持续时间内累积。如果设备的结果值高于或低于 (`comparisonOperator`) 与指定百分位数关联的值，则认为设备与行为相符。否则，该设备违反了该行为。

百分位数表示所有考虑的测量值中低于相关值的百分比。例如，如果与“p90”（第 90 个百分位数）相关的值是 123，则 90% 的测量值都低于 123。

`durationSeconds`

使用此参数为具有时间维度的条件（例如，`NUM_MESSAGES_SENT`）指定评估行为的时间段。对于 `statisticalThreshhold` 指标比较，这是一个时间段，期间将收集所有设备的测量值以确定 `statisticalThreshold` 值，然后为每个设备确定其行为在比较中的排名方式。

ML Detect

`ML Detect confidence`

ML Detect 支持三个置信级别：`High`、`Medium` 和 `Low`。`High` 置信度意味着异常行为评估的敏感度较低，同时告警数量往往较少,`Medium` 置信度意味着中等灵敏度，`Low` 置信度意味着高灵敏度和通常更高的告警数量。

ML Detect

通过机器学习 Detect (ML Detect)，您可以创建安全配置文件，使用机器学习通过基于历史设备数据自动创建模型来学习预期的设备行为，并将这些配置文件分配给一组设备或您机群中的所有设备。AWS IoT Device Defender 然后会使用 ML 模型识别异常并触发告警。

有关 ML Detect 入门的信息，请参阅 ML Detect 指南。

本章包含以下部分：

- ML Detect 的使用案例
- ML Detect 的工作原理
- 最低要求
- 限制
- 在告警中标记误报和其它验证状态
- 受支持的指标
- 服务限额
- ML Detect CLI 命令
- ML Detect API
- 暂停或删除 ML Detect 安全配置文件

ML Detect 的使用案例

当难以设置设备的预期行为时，您可以使用 ML Detect 来监控您的实例集设备。例如，要监控断开连接数量指标，可能不清楚什么是可接受的阈值。在这种情况下，您可以启用 ML Detect 功能，根据设备报告的历史数据来识别异常断开连接指标数据点。

ML Detect 的另一个使用案例是监控随着时间的推移动态变化的设备行为。ML Detect 根据设备中不断变更的数据模式，定期了解动态预期的设备行为。例如，发送的设备消息量在工作日和周末之间可能会有所不同，而 ML Detect 将了解此动态行为。

ML Detect 的工作原理

使用 ML Detect 功能，您可以创建行为以跨 6 个云端指标和 7 个设备端指标识别运营和安全异常。在初始模型训练期之后，ML Detect 会根据最后 14 天的数据每天刷新模型。它使用 ML 模型监控这些指标的数据点，并在检测到异常时触发告警。

如果将安全配置文件附加到具有类似预期行为的设备集合，则 ML Detect 将发挥出最大的作用。例如，如果您的某些设备在客户家中使用，而其他设备在企业办公室中使用，则这两个组之间的设备行为模式可能会有很大差异。您可以将设备组织到家用设备事物组和办公设备事物组。为了获得最佳异常检测效果，请将每个事物组附加到单独的 ML Detect 安全配置文件。

虽然 ML Detect 将构建初始模型，但在随后的 14 天期间内，每个指标需要 14 天和至少 25000 个数据点才能生成模型。之后，它每天都会更新模型，达到最少数量的指标数据点。如果未满足最低要求，ML Detect 会在第二天尝试构建模型，并在接下来的 30 天内每天重试，然后停止模型以进行评估。

最低要求

对于训练和创建初始 ML 模型，ML Detect 具有以下最低要求。

最低训练时间

构建初始模型需要 14 天的时间。之后，模型每天都会使用 14 天后续周期的指标数据进行刷新。

最低总数据点

在过去 14 天内，构建 ML 模型所需的最低数据点为每个指标 25000 个数据点。对于模型的持续训练和刷新，ML Detect 要求受监控设备满足最少的数据点。它大致相当于以下设置：

- 60 台设备连接并在 AWS IoT 上每隔 45 分钟执行一次活动。
- 40 台设备则每隔 30 分钟一次。
- 15 台设备则每隔 10 分钟一次。
- 7 台设备则每隔 5 分钟一次。

设备组目标

为收集数据，您必须在安全配置文件的目标事物组中包含事物。

创建初始模型后，ML 模型每天刷新，并在 14 天的后续周期内至少需要 25000 个数据点。

限制

您可以将 ML Detect 与有关以下云端指标的维度一起使用：

- 授权失败 (aws:num-authorization-failures)
- 已收到的消息 (aws:num-messages-received)
- 已发送的消息 (aws:num-messages-sent)
- 消息大小 (aws:message-byte-size)

ML Detect 不支持以下指标。

ML Detect 不支持的云端指标：

- 源 IP (aws:source-ip-address)

ML Detect 不支持的设备端指标：

- 目标 IP (aws:destination-ip-addresses)
- 侦听 TCP 端口 (aws:listening-tcp-ports)
- 侦听 UDP 端口 (aws:listening-udp-ports)

自定义指标仅支持 number（数值）类型。

在告警中标记误报和其它验证状态

如果您通过调查验证 ML 检测告警是否为误报，可以将告警的验证状态设置为误报。这可以帮助您和您的团队识别不必回应的告警。您还可以将告警标记为 “真”、“良性”或“未知”。

您可以通过 AWS IoT Device Defender 控制台或者使用 PutVerificationStateOnViolation API 操作。

受支持的指标

您可以将以下云端指标与 ML Detect 结合使用：

- 授权失败 (aws:num-authorization-failures)

- 连接尝试 (aws:num-connection-attempts)
- 断开连接 (aws:num-disconnects)
- 消息大小 (aws:message-byte-size)
- 已发送的消息 (aws:num-messages-sent)
- 已收到的消息 (aws:num-messages-received)

您可以将以下设备端指标与 ML Detect 结合使用：

- 输出字节数 (aws:all-bytes-out)
- 字节数 (aws:all-bytes-in)
- 侦听 TCP 端口计数 (aws:num-listening-tcp-ports)
- 侦听 UDP 端口计数 (aws:num-listening-udp-ports)
- 输出数据包数 (aws:all-packets-out)
- 数据包数 (aws:all-packets-in)
- 已建立的 TCP 连接计数 (aws:num-established-tcp-connections)

服务限额

有关 ML Detect 服务配额和限制的信息，请参阅 AWS IoT Device Defender 端点和配额。

ML Detect CLI 命令

您可以使用以下 CLI 命令来创建和管理 ML Detect。

- 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

以下 API 可用于创建和管理 ML Detect 安全配置文件。

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

暂停或删除 ML Detect 安全配置文件

您可以暂停 ML Detect 安全配置文件以暂时停止监控设备行为，也可以删除 ML Detect 安全配置文件以长时间停止监控设备行为。

使用控制台暂停 ML Detect 安全配置文件

要使用控制台暂停 ML Detect 安全配置文件，您必须首先有一个空的事物组。要创建空的事物组，请参阅《AWS IoT Core 开发人员指南》中的静态事物组。如果已创建空事物组，则将空事物组设置为 ML Detect 安全配置文件的目标。

> **Note**
>
> 您需要在 30 天内将安全配置文件的目标设置回具有设备的设备组，否则您将无法重新激活安全配置文件。

使用控制台删除 ML Detect 安全配置文件

若要删除安全配置文件，请按照下列步骤操作：

1. 在 AWS IoT 控制台导航到边栏，然后选择 Defend（防护）部分。
2. 在 Defend（防护）中，选择 Detect（检测）然后选择 Security Profiles（安全配置文件）。
3. 选择要删除的 ML Detect 安全配置文件。
4. 选择 Actions（操作），然后从选项中选择 Delete（删除）。

> **Note**
>
> 删除 ML Detect 安全配置文件后，您将无法重新激活安全配置文件。

使用 CLI 暂停 ML Detect 安全配置文件

要使用 CLI 暂停 ML Detect 安全配置文件，请使用 `detach-security-security-profile` 命令：

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

> **Note**
>
> 此选项只在 AWS CLI 中可用。与控制台工作流程类似，您需要在 30 天内将安全配置文件的目标设置回具有设备的设备组，否则您将无法重新激活安全配置文件。要将安全配置文件附加到设备组，请使用 `attach-security-profile` 命令。

使用 CLI 删除 ML Detect 安全配置文件

您可以通过使用如下 `delete-security-profile` 命令删除安全配置文件：

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

> **Note**
>
> 删除 ML Detect 安全配置文件后，您将无法重新激活安全配置文件。

自定义指标

借助 AWS IoT Device Defender 自定义指标，您可以定义和监控实例集或使用案例独有的指标，例如连接到 Wi-Fi 网关的设备数量、电池的充电量或智能插头的电源周期次数。自定义指标行为在安全配置文件中定义，其为设备组（事物组）或所有设备指定了预期行为。您可以通过设置告警来监控行为，该告警可用来检测和响应特定于设备的问题。

本章包含以下部分：

- 如何在控制台中使用自定义指标
- 如何使用 CLI 中的自定义指标
- 自定义指标 CLI 命令
- 自定义指标 API

如何在控制台中使用自定义指标

教程

- AWS IoT Device Defender 代理 SDK (Python)
- 创建自定义指标并将其添加到安全配置文件
- 查看自定义指标详细信息
- 更新自定义指标
- 删除自定义指标

AWS IoT Device Defender 代理 SDK (Python)

首先，请下载 AWS IoT Device Defender 代理 SDK (Python) 示例代理。代理将收集指标并发布报告。设备端指标发布后，您可以查看正在收集的指标并确定设置告警的阈值。有关设置设备代理的说明，请参阅 AWS IoT Device Defender 代理 SDK (Python) 自述文件。有关更多信息，请参阅 AWS IoT Device Defender 代理 SDK (Python)。

创建自定义指标并将其添加到安全配置文件

以下流程介绍了如何使用控制台创建自定义指标。

1. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Metrics（指标）。
2. 在 Custom metrics（自定义指标）页面上，选择 Create（创建）。

3. 在 Create custom metric（创建自定义指标）页面上，执行以下操作。

   1. 在 Name（名称）项下，输入自定义指标的名称。创建自定义指标之后，您无法修改此名称。
   2. 在 Display name (optional)（显示名称（可选））项下，您可以输入自定义指标的友好名称。它不需要是唯一的，并且可以在创建后进行修改。
   3. 在 Type（类型）中，选择要监控的指标类型。指标类型包括 string-list（字符串列表）、ip-address-list（IP 地址列表）、number-list（编号列表）和 number（编号）。创建后无法修改类型。

      > **Note**
      >
      > ML Detect 仅允许 number（数值）类型。

   4. 在 Tags（标签）项下，您可以选择要与资源关联的标签。

   完成后，选择 Confirm（确认）。
4. 创建自定义指标后，将会显示 Custom metrics（自定义指标）页面，您可以在其中查看新创建的自定义指标。
5. 接下来，您需要将自定义指标添加到安全配置文件。在 AWS IoT 控制台中的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Security profiles（安全配置文件）。
6. 选择您要添加自定义指标的安全配置文件。
7. 选择 Actions（操作）和 Edit（编辑）。
8. 选择 Addtional Metrics to retain（要保留的其它指标），然后选择您的自定义指标。在以下页面选择 Next（下一步），直到您到达 Confirm（确认）页面。选择 Save（保存）和 Continue（继续）。成功添加自定义指标后，将会显示安全配置文件详细信息页面。

   > **Note**
   >
   > 当任何指标值为负数时，百分位数统计数据不可用于指标。

查看自定义指标详细信息

以下流程介绍了如何在控制台中查看自定义指标的详细信息。

1. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Metrics（指标）。

2. 选择您想要查看详细信息的自定义指标的 Metric name（指标名称）。

更新自定义指标

以下流程说明如何使用控制台更新自定义指标。

1. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Metrics（指标）。
2. 选择要更新的自定义指标旁边的选项按钮。然后，在 Actions（操作）中，选择 Edit（编辑）。
3. 在 Update custom metrics（更新自定义指标）页面上，您可以编辑显示名称及删除或添加标签。
4. 完成后，选择 Update（更新）。Custom metrics（自定义指标）页面。

删除自定义指标

以下流程介绍了如何使用控制台删除自定义指标。

1. 首先，从引用的任何安全配置文件中删除您的自定义指标。您可以在自定义指标详细信息页面上查看哪些安全配置文件包含了您的自定义指标。在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Metrics（指标）。
2. 选择您要删除的自定义指标。从自定义指标详细新页面上的 Security Profiles（安全配置文件）下列示的任何安全配置文件中删除自定义指标。
3. 在 AWS IoT 控制台的导航窗格中，展开 Defend（防护），然后选择 Detect（检测）、Metrics（指标）。
4. 选择要删除的自定义指标旁边的选项按钮。然后，对于 Actions（操作），选择 Delete（删除）。
5. 在 Are you sure you want to delete custom metric?（是否确定要删除自定义指标）消息上，选择 Delete custom metric（删除自定义指标）。

> **Warning**
>
> 删除自定义指标后，将丢失与该指标关联的所有数据。此操作无法撤消。

如何使用 CLI 中的自定义指标

教程

- AWS IoT Device Defender 代理 SDK (Python)

- 创建自定义指标并将其添加到安全配置文件
- 查看自定义指标详细信息
- 更新自定义指标
- 删除自定义指标

AWS IoT Device Defender 代理 SDK (Python)

首先，请下载 AWS IoT Device Defender 代理 SDK (Python) 示例代理。代理将收集指标并发布报告。发布设备端指标后，您可以查看正在收集的指标并确定设置告警的阈值。有关设置设备代理的说明，请参阅 AWS IoT Device Defender 代理 SDK (Python) 自述文件。有关更多信息，请参阅 AWS IoT Device Defender 代理 SDK (Python)。

创建自定义指标并将其添加到安全配置文件

以下流程介绍如何从 CLI 创建自定义指标并将其添加到安全配置文件。

1. 使用 `create-custom-metric` 命令创建自定义指标。以下示例创建用于测量电池百分比的自定义指标。

```
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" \
```

输出：

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

2. 创建自定义量度指标后，您可以使用 `update-security-profile` 将自定义指标添加到现有配置文件或使用`create-security-profile` 创建新的配置文件以便添加自定义指标。在这里，我们新建一个名为 *batteryUsage* 的安全配置文件，以便将我们的新 *batteryPercentage* 自定义指标添加到其中。我们还添加了一个名为 *cellularBandwidth* 的 Rules Detect 指标。

```
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
```

输出：

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

> **Note**
>
> 当任何指标值为负数时，百分位数统计数据不可用于指标。

查看自定义指标详细信息

以下流程介绍如何从 CLI 查看自定义指标的详细信息。

- 使用 list-custom-metrics 命令查看所有自定义指标。

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

此命令的输出如下所示。

```
{
    "metricNames": [
        "batteryPercentage"
```

```
    ]
}
```

更新自定义指标

以下流程介绍如何从 CLI 更新自定义指标。

- 使用 update-custom-metric 命令更新自定义指标。以下示例将更新 display-name。

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

此命令的输出如下所示。

```
{
    "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"
}
```

删除自定义指标

以下流程介绍如何从 CLI 中删除自定义指标。

1. 要删除自定义指标，请先将其从附加到的任何安全配置文件中删除。使用 list-security-profiles 命令查看具有特定自定义指标的安全配置文件。
2. 要从安全配置文件中删除自定义指标，请使用 update-security-profiles 命令。输入要保留的所有信息，但排除自定义指标。

```
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}}]"
```

此命令的输出如下所示。

```
{
  "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. 分离自定义指标后，使用 delete-custom-metric 命令删除自定义指标。

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

此命令的输出如下所示

```
HTTP 200
```

自定义指标 CLI 命令

可以使用以下 CLI 命令来创建和管理自定义指标。

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

- list-security-profiles

自定义指标 API

以下 API 可用于创建和管理自定义指标。

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

设备端指标

创建安全配置文件时，您可以通过为 IoT 设备生成的指标配置行为和阈值来指定 IoT 设备的预期行为。以下是设备端指标，它们是来自您在设备上安装的代理的指标。

输出字节数 (`aws:all-bytes-out`)

给定时间段内从设备发出的出站字节数量。

使用此指标指定给定时间段内设备应该发送的最大或最小出站流量（以字节为单位）。

兼容：Rules Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：字节

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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 使用 **statisticalThreshold** 的示例

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

Example 使用 ML Detect 的示例

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

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

字节数 (**aws:all-bytes-in**)

给定时间段内发往设备的入站字节数量。

使用此指标指定给定时间段内设备应该接收的最大或最小入站流量（以字节为单位）。

兼容：Rules Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：字节

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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 使用 **statisticalThreshold** 的示例

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-bytes-in",
```

```
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example 使用 ML Detect 的示例

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

侦听 TCP 端口计数 (**aws:num-listening-tcp-ports**)

设备正在侦听的 TCP 端口数量。

使用此指标指定每个设备应该监控的最大 TCP 端口数量。

兼容：Rules Detect | ML Detect

单位：失败次数

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：失败次数

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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 使用 **statisticalThreshold** 的示例

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

Example 使用 ML Detect 的示例

```
{
  "name": "Max TCP Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
```

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

侦听 UDP 端口计数 (**aws:num-listening-udp-ports**)

设备正在侦听的 UDP 端口数量。

使用此指标指定每个设备应该侦听的最大 UDP 端口数量。

兼容：Rules Detect | ML Detect

单位：失败次数

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：失败次数

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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 使用 **statisticalThreshold** 的示例

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

Example 使用 ML Detect 的示例

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

输出数据包数 (**aws:all-packets-out**)

给定时间段内从设备发出的出站数据包数量。

使用此指标指定给定时间段内设备应该发送的最大或最小出站总流量。

兼容：Rules Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：数据包

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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 使用 **`statisticalThreshold`** 的示例

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

Example 使用 ML Detect 的示例

```
{
  "name": "Outbound sent ML behavior",
```

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

数据包数 (**aws:all-packets-in**)

给定时间段内发往设备的入站数据包数量。

使用此指标指定给定时间段内设备应该接收的最大或最小入站总流量。

兼容：Rule Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：数据包

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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

使用 statisticalThreshold 的示例

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

Example 使用 ML Detect 的示例

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

目标 IP (**aws:destination-ip-addresses**)

一组目标 IP 地址。

使用此指标指定一组允许（以前称为列入白名单）或拒绝（以前称为列入黑名单）的无类域间路由 (CIDR)，每个设备必须或不得通过它们连接到 AWS IoT。

兼容：Rule Detect

运算符：in-cidr-set | not-in-cidr-set

值：CIDR 列表

单位：n/a

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
}
```

侦听 TCP 端口 (**aws:listening-tcp-ports**)

设备正在侦听的 TCP 端口。

使用此指标指定一组允许（以前称为列入白名单）或拒绝（以前称为列入黑名单）的 TCP 端口，每个设备必须或不得通过它们进行侦听。

兼容：Rule Detect

运算符：in-port-set | not-in-port-set

值：端口列表

单位：n/a

Example

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
```

```
    "value": {
      "ports": [ 443, 80 ]
    }
  },
  "suppressAlerts": true
}
```

侦听 UDP 端口 (**aws:listening-udp-ports**)

设备正在侦听的 UDP 端口。

使用此指标指定一组允许（以前称为列入白名单）或拒绝（以前称为列入黑名单）的 UDP 端口，每个设备必须或不得通过它们进行侦听。

兼容：Rule Detect

运算符：in-port-set | not-in-port-set

值：端口列表

单位：n/a

Example

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

已建立的 TCP 连接计数 (**aws:num-established-tcp-connections**)

设备的 TCP 连接数。

使用此指标指定每个设备应该具有的最大或最小活动 TCP 连接数量（所有 TCP 状态）。

兼容：Rules Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：连接

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 使用 **statisticalThreshold** 的示例

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

Example 使用 ML Detect 的示例

```
{
```

```
  "name": "Connection count ML behavior",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

设备指标文档规范

整体结构

长名称	短名称	必需	Type	约束	注意事项
header	hed	Y	对象		格式正确的报告所必需的完整数据块。
指标	met	Y	对象		报告可以同时包含两个或至少一个 `metrics` 或者 `custom_metrics` 数据块。
custom_metrics	cmet	Y	对象		报告可以同时包含两个或至少一个 `metrics` 或者 `custom_metrics` 数据块。

标头数据块

长名称	短名称	必需	Type	约束	注意事项
report_id	rid	Y	整数		单调递增值。推荐采用纪元时间戳。
version	v	Y	字符串	Major.Minor	添加字段时为次要增量。如果指标移除，则为主要增量。

指标数据块：

TCP 连接

长名称	短名称	父元素	必需	Type	约束	注意事项
tcp_connections	tc	指标	否	对象		
established_connections	ec	tcp_connections	否	对象		已建立 TCP 状态
connections	cs	established_connections	否	List<Object>		
remote_addr	rad	connections	Y	数字	ip:port	IP 可能是 IPv6 或 IPv4
local_port	lp	connections	否	数字	>= 0	

长名称	短名称	父元素	必需	Type	约束	注意事项
local_interface	li	connections	否	字符串		接口名称
total	t	established_connections	否	数字	>= 0	已建立的连接数

侦听 TCP 端口

长名称	短名称	父元素	必需	Type	约束	注意事项
listening_tcp_ports	tp	指标	否	对象		
ports	pts	listening_tcp_ports	否	List<Object>	> 0	
port	pt	ports	否	数字	> 0	端口应该是大于 0 的数字
interface	if	ports	否	字符串		接口名称
total	t	listening_tcp_ports	否	数字	>= 0	

侦听 UDP 端口

长名称	短名称	父元素	必需	Type	约束	注意事项
listening_udp_ports	up	指标	否	对象		
ports	pts	listening_udp_ports	否	List<Port>	> 0	

长名称	短名称	父元素	必需	Type	约束	注意事项
port	pt	ports	否	数字	> 0	端口应该是大于 0 的数字
interface	if	ports	否	字符串		接口名称
total	t	listening_udp_ports	否	数字	>= 0	

网络统计数据

长名称	短名称	父元素	必需	Type	约束	注意事项
network_stats	ns	metrics	否	对象		
bytes_in	bi	network_stats	否	数字	Delta Metric, >= 0	
bytes_out	bo	network_stats	否	数字	Delta Metric, >= 0	
packets_in	pi	network_stats	否	数字	Delta Metric, >= 0	
packets_out	po	network_stats	否	数字	Delta Metric, >= 0	

Example

以下 JSON 结构使用长名称。

```
{
  "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 使用短名称的 JSON 结构示例

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

从设备发送指标

AWS IoT Device Defender Detect 可以收集、聚合和监控 AWS IoT 设备生成的指标数据，以识别表现出异常行为的设备。本部分介绍如何将指标从设备发送到 AWS IoT Device Defender。

您必须在连接 AWS IoT 的设备或设备网关上安全部署 AWS IoT SDK 版本二，以收集设备端指标。查看在此处查看 SDK 的完整列表。

您可以使用 AWS IoT Device Client 发布指标，因为它提供的单个代理涵盖了 AWS IoT Device Defender 和 AWS IoT Device Management 中的功能。这些功能包括任务、安全隧道、AWS IoT Device Defender 指标发布等。

您可以将设备端指标发布到 AWS IoT 中的预留主题，便于 AWS IoT Device Defender 进行收集和评估。

使用 AWS IoT Device Client 发布指标

要安装 AWS IoT Device Client，您可以从 Github 下载。在要收集设备端数据的设备上安装 AWS IoT 后，您必须进行配置以将设备端指标发送到 AWS IoT Device Defender。确认 AWS IoT Device Client 配置文件在 `device-defender` 部分设置了以下参数：

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

```
    }
```

> **Warning**
>
> 时间间隔至少应设置为 300 秒。如果将时间间隔设置为小于 300 秒，则您的指标数据可能会被节流。

更新配置后，您可以在 AWS IoT Device Defender 控制台创建安全配置文件和行为来监控设备发布到云的指标。您可以依次选择 Defend（防护）、Detect（检测）和 Metrics（指标），从而在 AWS IoT Core 控制台中发布指标。

云端指标

创建安全配置文件时，您可以通过为 IoT 设备生成的指标配置行为和阈值来指定 IoT 设备的预期行为。以下是云端指标，来自 AWS IoT。

消息大小 (aws:message-byte-size)

消息中的字节数。使用此指标指定从设备传输到 AWS IoT 的每条消息的最大或最小大小（以字节为单位）。

兼容：Rules Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：字节

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 使用 **statisticalThreshold** 的示例

```
{

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

Example 使用 ML Detect 的示例

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

如果设备在连续三个五分钟时段内，传输的消息累积大小超过针对该安全配置文件行为报告的所有其他设备累积大小的 90%，则会发出告警。

已发送的消息 (aws:num-messages-sent)

设备在给定时间段内发送的消息数量。

使用此指标指定给定时间段内 AWS IoT 与各个设备之间可发送的最大或最小消息数量。

兼容：Rules Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：消息

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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 使用 **`statisticalThreshold`** 的示例

```
{

  "name": "Out bound message rate",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
```

```
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example 使用 ML Detect 的示例

```
{
  "name": "Messages sent ML behavior",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

已收到的消息 (aws:num-messages-received)

设备在给定时间段内接收的消息数量。

使用此指标指定给定时间段内 AWS IoT 与各个设备之间可接收的最大或最小消息数量。

兼容：Rules Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：消息

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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 使用 **statisticalThreshold** 的示例

```
{
  "name": "In bound message rate",
  "metric": "aws:num-messages-received",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example 使用 ML Detect 的示例

```
{
  "name": "Messages received ML behavior",
  "metric": "aws:num-messages-received",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
```

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

授权失败 (aws:num-authorization-failures)

使用此指标指定给定时间段内每个设备允许的最大授权失败次数。如果从设备发往 AWS IoT 的请求遭到拒绝（例如，设备试图发布到某个主题，但没有足够的权限），则会发生授权失败。

兼容：Rules Detect | ML Detect

单位：失败次数

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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 使用 **`statisticalThreshold`** 的示例

```
{
  "name": "Authorization Failures",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "comparisonOperator": "less-than-equals",
```

```
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example 使用 ML Detect 的示例

```
{
  "name": "Authorization failures ML behavior",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

源 IP (aws:source-ip-address)

设备连接到 AWS IoT 的 IP 地址。

使用此指标指定一组允许（以前称为列入白名单）或拒绝（以前称为列入黑名单）的无类域间路由 (CIDR)，每个设备必须或不得通过它们连接到 AWS IoT。

兼容：Rule Detect

运算符：in-cidr-set | not-in-cidr-set

值：CIDR 列表

单位：n/a

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
}
```

连接尝试 (aws:num-connection-attempts)

设备在给定时间段内尝试建立连接的次数。

使用此指标指定每个设备尝试建立连接的最大或最小次数。成功和失败的尝试都会计算在内。

兼容：Rules Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：连接尝试次数

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

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 使用 **statisticalThreshold** 的示例

```
{
  "name": "Connection Attempts",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example 使用 ML Detect 的示例

```
{
  "name": "Connection attempts ML behavior",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": false
}
```

断开连接 (aws:num-disconnects)

设备在给定时间段内与 AWS IoT 断开连接的次数。

使用此指标来指定设备在给定时间段内与 AWS IoT 断开连接的最大或最小次数。

兼容：Rules Detect | ML Detect

运算符：less-than | less-than-equals | greater-than | greater-than-equals

值：非负整数

单位：连接断开次数

持续时间：非负整数。有效值为 300、600、900、1800 或 3600 秒。

Example

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example 使用 **`statisticalThreshold`** 的示例

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example 使用 ML Detect 的示例

```
{
  "name": "Disconnects ML behavior",
```

```
  "metric": "aws:num-disconnects",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

断开连接持续时间（aws:disconnect-duration）

设备与 AWS IoT 断开连接的持续时间。

使用此指标指定设备与 AWS IoT 保持断开连接的最大持续时间。

兼容：Rule Detect

运算符：less-than | less-than-equals

值：非负整数（分钟）

Example

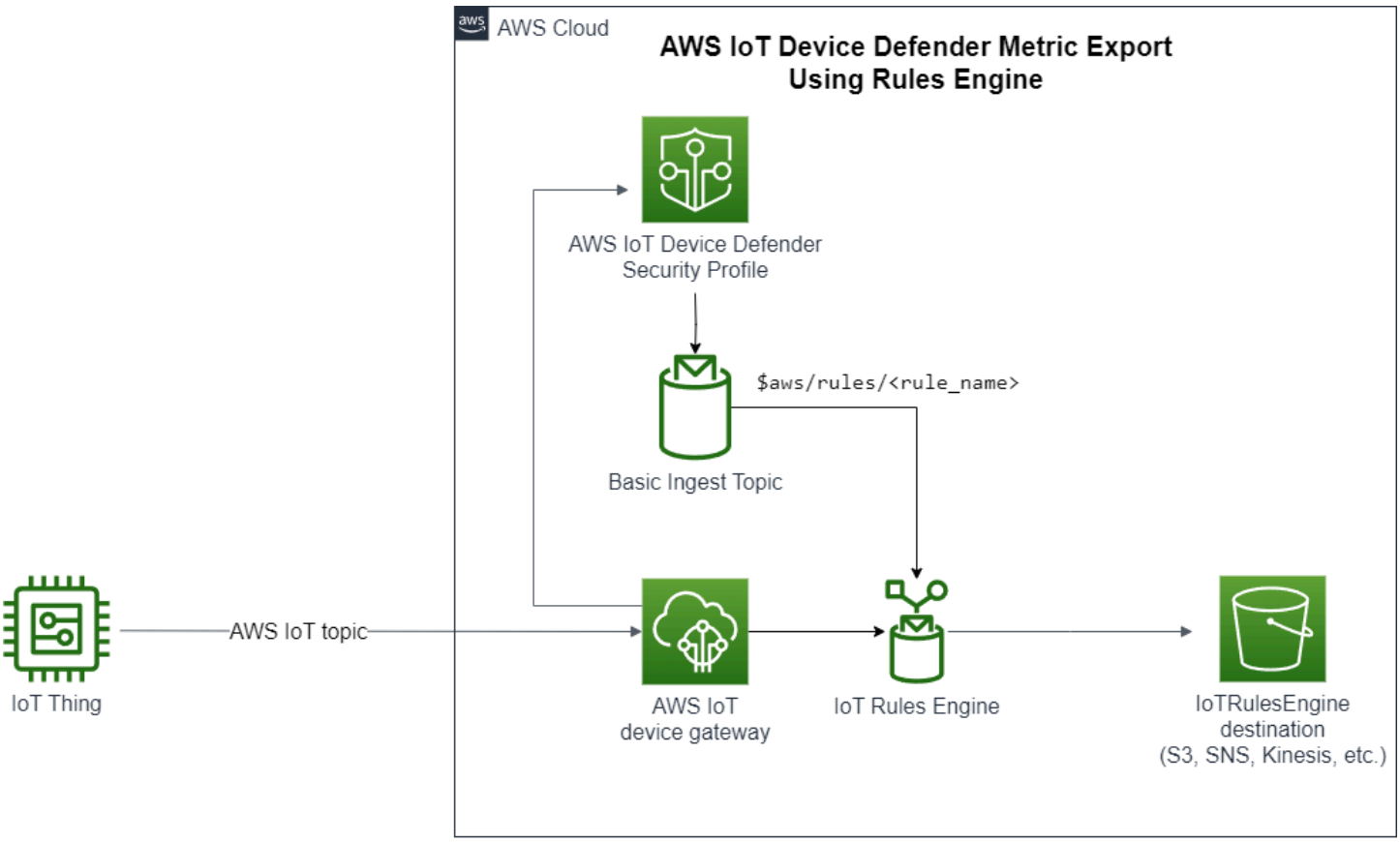
```
{
"name": "DisconnectDuration",
  "metric": "aws:disconnect-duration",
  "criteria": {
"comparisonOperator": "less-than-equals",
    "value": {
"count": 5
    }
  },
  "suppressAlerts": true
}
```

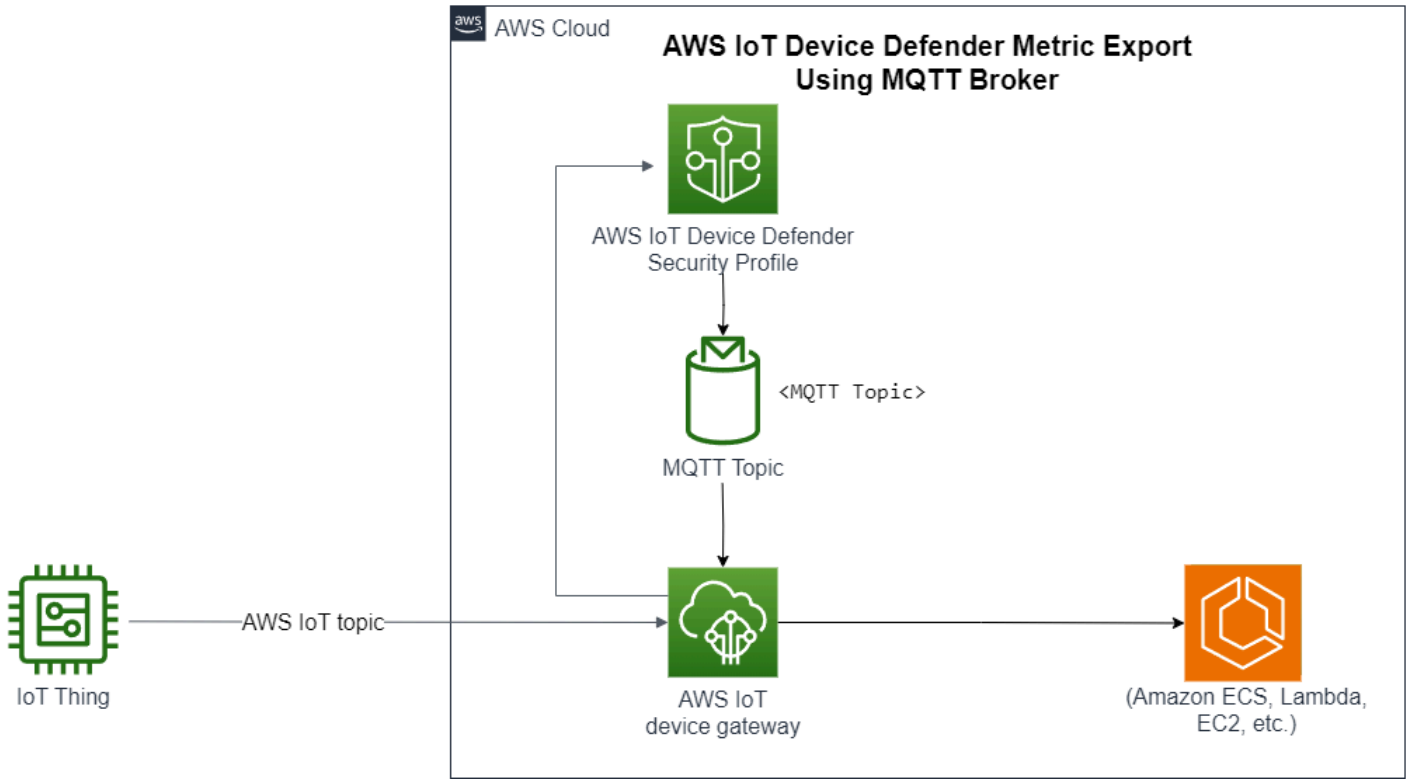
Detect 指标导出

使用指标导出，您可以从 AWS IoT Device Defender 中导出云端、设备端或自定义指标，并将它们发布到您配置的 MQTT 主题。此功能支持批量导出 Detect 指标，这不仅可以提高数据报告和分析的效

率，还有助于控制成本。您可以选择您的 MQTT 主题作为 AWS IoT 规则基本摄取主题，也可以创建和订阅自己的 MQTT 主题。使用 AWS IoT Device Defender 控制台、API 或 CLI 配置指标导出。此功能在所有提供 AWS IoT Device Defender 的 AWS 区域中均可用。

下图显示了如何配置 AWS IoT Device Defender 来导出指标。第一个图表演示了如何对基本摄取主题配置导出指标。然后，您可以将导出的指标路由到 AWS IoT 规则支持的不同目标。第二个图表显示了如何配置 AWS IoT Device Defender 将数据发布到 MQTT 主题。然后，MQTT 客户端订阅该主题。您可以在订阅相同 MQTT 主题的 Amazon Elastic Container Service、Lambda 或 Amazon EC2 实例上的容器中运行 MQTT 客户端。每当 AWS IoT Device Defender 发布数据时，MQTT 客户端就会收到并处理数据。有关更多信息，请参阅 MQTT 主题。


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.)

检测指标导出的工作原理

设置安全配置文件时，您可以选择要导出的指标并指定 MQTT 主题。还可以配置一个 IAM 角色，该角色授予 AWS IoT Device Defender Detect 向已配置的 MQTT 主题发布消息所需的权限。您可以配置 AWS IoT 规则基本注入 MQTT 主题并将导出的指标发送到 AWS IoT 规则支持的目标。有关设置和配置 AWS IoT 规则的说明，请参阅《AWS IoT 开发人员指南》中的 AWS IoT 规则。

AWS IoT Device Defender Detect 对每个已配置指标的指标值进行批处理，并定期将其发布到已配置的 MQTT 主题。除了消息字节大小和总字节大小之外，云端指标是通过对批处理持续时间的指标值求和来聚合的。不聚合自定义指标和设备端指标。对于消息字节大小，导出值是批处理持续时间内的最小字节大小、最大字节大小和总字节大小。对于断开连接持续时间，导出值是所有被跟踪设备的断开连接持续时间（以秒为单位）。这种情况每隔一个小时发生一次，对于连接或断开连接事件也是如此。对于连接的设备或连接事件，该值为零。有关云端指标、设备端指标和自定义指标的更多信息，请参阅《AWS IoT Device Defender 开发人员指南》中的以下主题：

- 自定义指标
- 云端指标
- 设备端指标

您可以使用 AWS IoT 规则将批处理指标导出到不同的目标。有关支持的目标列表，请参阅 AWS IoT 规则操作。要将批处理导出消息中的个别指标发送到受支持的目标，请对 AWS IoT 规则操作使用 batchMode 选项。如果您首选的 AWS IoT 规则目标缺少 `batchMode` 支持，您仍然可以通过使用 Lambda 或 Kinesis Data Streams 等中间操作在批处理消息中发送个别指标。

指标导出模式

有关批量指标导出数据，请参阅以下模式。

```
{
 "version": "1.0",
 "metrics": [
 {
 "name": "{metricName}",
 "thing": "{thingName}",
 "value": {
 # a list of Classless Inter-Domain Routings (CIDR) specifying metric
# source-ip-address and destination-ip-address
 "cidrs": ["string"],
```

```
 # a single metric value for cloud/device metrics
 "count": number,
 # a single metric value for custom metric
 "number": number,
 # a list of numbers for custom metrics
 "numbers": [number],
 # a list of ports for cloud/device metrics
 "ports": [number],
 # a list of strings for custom metrics
 "strings": ["string"]
 },
 # In some rare cases we may send multiple values for the same thing, metric and
 timestamp.
 # When there are multiple values, please use the value with highest version number
 # and discard other values.
 "version": number,
 # For cloud-side metrics, this is the time when AWS IoT Device Defender Detect
 aggregates the
 # metrics data received from AWS IoT.
 # For device-side and custom metrics, this is the time at which the metrics data
 # is reported by the devices.
 "timestamp": number,
 # The dimension parameters are optional. It's set only if
 # the metrics are configured with a dimension in the security profile.
 "dimension": {
 "name": "{dimensionName}",
 "operator": "{dimensionOperator}"
 }
 }
 ]
}
```

Detect 指标导出定价

当您将云端、设备端或自定义指标发布到您配置的 MQTT 主题时，您不会为导出过程的这一步产生费用。但是，在后续步骤中，当您使用规则引擎或消息收发将已发布的指标传输到您选择的目标时，将根据您选择的传输方式产生费用。AWS IoT Device Defender 将批处理指标作为单条消息发布到 MQTT 主题，其中包含多个设备的指标数据，这有助于控制成本。有关定价的更多信息，请参阅 AWS Pricing Calculator。

权限

本节包含有关如何设置管理 AWS IoT Device Defender Detect 指标导出所需的 IAM 角色和策略的信息。有关更多信息，请参阅 IAM 用户指南。

授予 AWS IoT Device Defender Detect 向 MQTT 主题发布消息的权限

如果在 CreateSecurityProfile 中启用指标导出，则必须为 IAM 角色指定两个策略：一个权限策略和一个信任策略。权限策略授予 AWS IoT Device Defender 向 MQTT 主题发布消息（包含指标）的权限。信任策略授予 AWS IoT Device Defender 代入所需角色的权限。

权限策略

```
{
  "Version":"2012-10-17",
  "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "iot:Publish"
        ],
        "Resource":[
          "arn:aws:iot:region:account-id:topic/your-topic-name"
        ]
    }
  ]
}
```

信任策略

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
```

```
}
```

传递角色策略

您还需要附加至 IAM 用户的 IAM 权限策略，允许该用户传递角色。请参阅向用户授予权限以将角色传递给 AWS 服务

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
          "iam:GetRole",
          "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/Role_To_Pass"
    }
  ]
}
```

在 AWS IoT 控制台中设置 Detect 指标导出

在控制台中创建、查看和编辑包含指标导出的新安全配置文件。

先决条件

在您设置 Detect 指标导出之前，请确保您具有以下先决条件：

- IAM 角色。有关创建 IAM 角色的更多信息，请参阅《IAM 用户指南》中的创建 IAM 角色。
- 您可以使用 AWS（IAM）用户身份登录的具有正确权限的 AWS Identity and Access Management 账户。有关 AWS IoT Device Defender Detect 权限的更多信息，请参阅《AWS IoT Core 开发人员指南》中的权限。

创建包含指标导出的新安全配置文件（控制台）

要导出指标行为数据，首先将安全配置文件配置为包含指标导出。以下过程详细介绍了如何设置基于规则的安全配置文件，其中包含 Detect 指标导出。

创建包含指标导出的新安全配置文件

1. 打开AWS IoT控制台。在导航窗格上，依次展开安全、检测和安全配置文件。
2. 在创建安全配置文件中，选择创建基于规则的异常检测配置文件。
3. 要指定您的安全配置文件属性，请输入您的安全配置文件名称，然后对于目标，选择一组要进行异常检测的目标设备。（可选）包括描述和用于标记 AWS 资源的标签。选择下一步。
4. 对于指标，选择用于定义设备行为的指标。您可以定义行为阈值，以便在您的设备未达到行为预期时收到警报。
5. 要接收行为异常警报，请选择发送警报（定义指标行为），然后指定行为名称和条件。要在没有警报的情况下保留指标，请选择不要发送警报（保留指标）。选择下一步。
6. 要配置指标导出，请选择开启指标导出。
7. 输入用于将指标数据发布到 AWS IoT Core 的 MQTT 主题名称。选择一个 IAM 角色来授予 AWS IoT“AWS IoT:Publish”权限，以便它向配置的主题发布消息。选择要导出的指标，然后选择下一步。

   > **Note**
   >
   > 输入 MQTT 主题名称时，使用正斜杠表示层次信息。例如，`$AWS/rules/rule-name/`。

8. 要在设备违反设定行为时向您的 AWS 控制台发送警报，请选择或创建 Amazon SNS 主题和 IAM 角色。选择下一步。
9. 查看您的配置，然后选择下一步。

查看和编辑安全配置文件详细信息（控制台）

查看和编辑安全配置文件详细信息

1. 打开AWS IoT控制台。在导航窗格上，依次展开安全、检测和安全配置文件。
2. 选择您创建的要包含指标导出的安全配置文件，然后对于操作，选择编辑。
3. 在目标下，选择要编辑的目标设备组，然后选择下一步。
4. 要编辑指标行为配置，请选择提醒我（定义指标行为），然后定义满足指标行为时的条件。选择下一步。
5. 要关闭指标导出配置，请选择关闭导出指标。选择下一步。

6. 要将 Amazon SNS 配置为在设备违反设定行为时向您的 AWS IoT 控制台发送警报，请选择或创建 Amazon SNS 主题和 IAM 角色。选择下一步。
7. 查看您的配置，然后选择下一步。

创建安全配置文件来启用指标导出

使用 `create-security-profile` 命令创建您的安全配置文件并启用指标导出。

创建包含指标导出的安全配置文件

1. 要启用指标导出并指示 Detect 是否需要导出相应的指标，请在 `Behavior` 和 `AdditionalMetricsToRetainV2` 中将 `exportMetric` 值设置为 True。
2. 包括 `MetricsExportConfig` 的值。这会指定指标导出所需的 MQTT 主题和角色 Amazon 资源名称（ARN）。

> **Note**
>
> 包含 `mqttTopic`，让 AWS IoT Device Defender Detect 可以发布消息。此角色 ARN 有权发布 MQTT 消息，之后 AWS IoT Device Defender Detect 可以代入该角色并代表您发布消息。

```
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
```

输出：

```
{
    "securityProfileName": "CreateSecurityProfileWithMetricsExport",
```

```
    "securityProfileArn": "arn:aws:iot:us-east-1:123456789012:securityprofile/
CreateSecurityProfileWithMetricsExport"
}
```

更新安全配置文件来启用指标导出（CLI）

使用 `update-security-profile` 命令更新现有的安全配置文件并启用指标导出。

更新安全配置文件来启用指标导出

1. 要启用指标导出并指示 Detect 是否需要导出相应的指标，请在 `Behavior` 和 `AdditionalMetricsToRetainV2` 中将 `exportMetric` 值设置为 True。
2. 包括 `MetricsExportConfig` 的值。这会指定指标导出所需的 MQTT 主题和角色 Amazon 资源名称（ARN）。

> **Note**
>
> 包含 `mqttTopic`，让 AWS IoT Device Defender Detect 可以发布消息。此角色 ARN 有权发布 MQTT 消息，之后 AWS IoT Device Defender Detect 可以代入该角色并代表您发布消息。

```
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
```

输出：

```
{
    "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"
    }
}
```

更新安全配置文件来关闭指标导出（CLI）

使用 update-security-profile 命令更新现有的安全配置文件并关闭指标导出。

更新安全配置文件来关闭指标导出

- 要更新您的安全配置文件并移除指标导出配置，请使用命令 --delete-metrics-export-config。

```
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
```

输出：

```
{
    "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"
}
```

有关更多信息，请参阅《AWS IoT 开发人员指南》中 Detect 命令。

指标导出 CLI 命令

您可以使用以下 CLI 命令来创建和管理 Detect 指标导出。

- CreateSecurityProfile
- UpdateSecurityProfile
- DescribeSecurityProfile

指标导出 API 操作

您可以使用以下 API 操作来创建和管理 Detect 指标导出。

- CreateSecurityProfile
- UpdateSecurityProfile
- DescribeSecurityProfile

使用维度确定安全配置文件中指标的作用域

维度是您可以定义的属性，用于获取安全配置文件中有关指标和行为的更精确数据。您通过提供用作筛选条件的值或模式来定义作用域。例如，您可以定义一个主题筛选条件维度，该维度仅将指标应用于与特定值（例如“data/bulb/+/activity”）匹配的 MQTT 主题。有关定义要在安全配置文件中使用的维度的信息，请参阅 CreateDimension。

维度值支持 MQTT 通配符。MQTT 通配符可帮助您同时订阅多个主题。有两种不同类型的通配符：单级 (+) 和多级 (#)。例如，维度值 `Data/bulb/+/activity` 创建一个和与 + 处于同一级别的所有主题匹配的订阅。维度值还支持 MQTT 客户端 ID 替换变量 ${iot:ClientId}。

TOPIC_FILTER 类型的维度与以下云端衡量指标集兼容：

- 授权失败次数
- 消息字节大小
- 已收到消息的数量
- 已发送消息的数量
- 源 IP 地址（仅适用于 Rules Detect）

如何在控制台中使用维度

创建维度并将其应用于安全配置文件行为

1. 打开AWS IoT控制台。在导航窗格中，依次展开安全和检测，然后选择安全配置文件。
2. 在安全配置文件页面上，选择创建安全配置文件，然后选择创建基于规则的异常检测配置文件。或者，要将某个维度应用于现有的基于规则的安全配置文件，请选择该安全配置文件并选择编辑。
3. 在指定安全配置文件属性页面上，输入安全配置文件的名称。

4. 选择要作为异常的目标的设备组。
5. 选择下一步。
6. 在配置指标行为页面上，在指标类型下选择云端指标维度之一。
7. 对于指标行为，选择发送警报（定义指标行为）以定义预期的指标行为。
8. 选择何时希望收到有关设备异常行为的通知。
9. 选择下一步。
10. 查看安全配置文件配置，然后选择创建。

查看您的警报

1. 打开AWS IoT控制台。在导航窗格中，依次展开安全和检测，然后选择警报。
2. 在事物名称列中，选择事物以查看有关导致警报的原因的信息。

查看和更新您的维度

1. 打开AWS IoT控制台。在导航窗格中，依次展开安全和检测，然后选择维度。
2. 选择维度，然后选择编辑。
3. 编辑维度并选择更新。

删除维度

1. 打开AWS IoT控制台。在导航窗格中，依次展开安全和检测，然后选择维度。
2. 在删除维度之前，必须删除引用该维度的指标行为。通过检查安全配置文件列，确认维度未附加到安全配置文件。如果维度附加到安全配置文件，请打开左侧的安全配置文件页面，然后编辑维度附加到的安全配置文件。然后，您可以继续删除该行为。如果要删除其它维度，请按照本节中的步骤操作。
3. 选择维度，然后选择删除。
4. 输入要确认的维度名称，然后选择删除。

如何在 AWS CLI CLI 上使用维度

创建维度并将其应用于安全配置文件行为

1. 首先创建维度，然后再将其附加到安全配置文件。使用 CreateDimension 命令创建维度：

```
aws iot create-dimension \
  --name TopicFilterForAuthMessages \
  --type TOPIC_FILTER \
  --string-values device/+/auth
```

此命令的输出如下所示：

```
{
    "arn": "arn:aws:iot:us-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
    "name": "TopicFilterForAuthMessages"
}
```

2. 使用 UpdateSecurityProfile 将维度添加到现有安全配置文件中，或使用 CreateSecurityProfile 将维度添加到新的安全配置文件中。在以下示例中，我们创建了一个新的安全配置文件，用于检查 TopicFilterForAuthMessages 的消息是否在 128 字节以下，并保留发送到非身份验证主题的消息数。

```
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\"}}]"
```

此命令的输出如下所示：

```
{
    "securityProfileArn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice",
    "securityProfileName": "ProfileForConnectedDevice"
}
```

您也可以从文件加载参数，而不是将其作为命令行参数值完全键入，以节省时间。有关更多信息，请参阅从文件加载 AWS CLI 参数。下面显示了扩展 JSON 格式的 behavior 参数：

```
[
  {
    "criteria": {
      "comparisonOperator": "less-than",
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "value": {
        "count": 128
      }
    },
    "metric": "aws:message-byte-size",
    "metricDimension": {
      "dimensionName:": "TopicFilterForAuthMessages"
    },
    "name": "CellularBandwidth"
  }
]
```

或者，通过将维度与 ML 结合使用来使用 CreateSecurityProfile，如以下示例所示：

```
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
```

查看具有某个维度的安全配置文件

- 使用 ListSecurityProfiles 命令查看具有特定维度的安全配置文件：

```
aws iot list-security-profiles \
  --dimension-name TopicFilterForAuthMessages
```

此命令的输出如下所示：

```
{
    "securityProfileIdentifiers": [
        {
            "name": "ProfileForConnectedDevice",
            "arn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice"
        }
    ]
}
```

更新维度

- 使用 UpdateDimension 命令更新维度：

```
aws iot update-dimension \
  --name TopicFilterForAuthMessages \
  --string-values device/${iot:ClientId}/auth
```

此命令的输出如下所示：

```
{
    "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"
}
```

删除维度

1. 要删除维度，请先将其从附加到的任何安全配置文件中分离。使用 ListSecurityProfiles 命令查看具有特定维度的安全配置文件。

2. 要从安全配置文件中删除维度，请使用 UpdateSecurityProfile 命令。输入要保留的所有信息，但排除维度：

```
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}}]"
```

此命令的输出如下所示：

```
{
  "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. 分离维度后，使用 DeleteDimension 命令删除维度：

```
aws iot delete-dimension \
  --name TopicFilterForAuthMessages
```

权限

本部分包含有关如何设置管理 AWS IoT Device Defender Detect 所需的 IAM 角色和策略的信息。有关更多信息，请参阅 IAM 用户指南。

授予 AWS IoT Device Defender Detect 向 SNS 主题发布告警的权限

如果在 CreateSecurityProfile 中使用 `alertTargets` 参数，则必须为 IAM 角色指定两个策略：一个权限策略和一个信任策略。权限策略授予 AWS IoT Device Defender 向 SNS 主题发布通知的权限。信任策略授予 AWS IoT Device Defender 代入所需角色的权限。

权限策略

```
{
  "Version":"2012-10-17",
  "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "sns:Publish"
        ],
        "Resource":[
          "arn:aws:sns:region:account-id:your-topic-name"
        ]
```

```
    }
  ]
}
```

信任策略

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

传递角色策略

您还需要附加至 IAM 用户的 IAM 权限策略，允许该用户传递角色。请参阅[向用户授予权限以将角色传递给 AWS 服务]

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
          "iam:GetRole",
          "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/Role_To_Pass"
    }
  ]
}
```

Detect 命令

您可以使用本部分中的 Detect 命令来配置 ML Detect 或规 Rules Detect 安全配置文件，以识别和监控可能表明设备受到危害的异常行为。

DetectMitigation 操作命令

启动和管理 Detect 执行
CancelDetectMitigationActionsTask
DescribeDetectMitigationActionsTask
ListDetectMitigationActionsTasks
StartDetectMitigationActionsTask
ListDetectMitigationActionsExecutions

维度操作命令

启动和管理维度执行
CreateDimension
DescribeDimension
ListDimensions
DeleteDimension
UpdateDimension

CustomMetric 操作命令

启动和管理 CustomMetric 执行
CreateCustomMetric
UpdateCustomMetric

启动和管理 CustomMetric 执行
DescribeCustomMetric
ListCustomMetrics
DeleteCustomMetric

安全配置文件操作命令

启动和管理安全配置文件执行
CreateSecurityProfile
AttachSecurityProfile
DetachSecurityProfile
DeleteSecurityProfile
DescribeSecurityProfile
ListTargetsForSecurityProfile
UpdateSecurityProfile
ValidateSecurityProfileBehaviors
ListSecurityProfilesForTarget

告警操作命令

管理告警和目标
ListActiveViolations
ListViolationEvents
PutVerificationStateOnViolation

ML Detect 操作命令

列出 ML 模型训练数据
GetBehaviorModelTrainingSummaries

如何使用 AWS IoT Device Defender detect

1. 可以只将 AWS IoT Device Defender Detect 与云端指标相结合使用，但如果计划使用设备报告的指标，必须首先在连接 AWS IoT 的设备或设备网关上部署 AWS IoT SDK。有关更多信息，请参阅从设备发送指标。
2. 在定义行为并创建警报之前，请先考虑查看设备生成的指标。AWS IoT 可以从设备收集指标，因此您可以首先识别账户中的一组设备或所有设备的常见或异常行为。使用 CreateSecurityProfile，但只指定您感兴趣的那些 `additionalMetricsToRetain`。此时不要指定 `behaviors`。

   使用 AWS IoT 控制台查看设备指标，以了解设备的典型行为。
3. 为安全配置文件创建一组行为。行为包含为账户中的一组设备或所有设备指定正常行为的指标。有关详细信息和示例，请参阅云端指标和 设备端指标。创建一组行为后，可以使用 ValidateSecurityProfileBehaviors 进行验证。
4. 使用 CreateSecurityProfile 操作创建包含行为的安全配置文件。当设备违反行为时，可以使用 `alertTargets` 参数向目标（SNS 主题）发送告警。（如果使用 SNS 发送告警，请注意系统将根据您AWS 账户的 SNS 主题配额统计数量。）大规模的违规项可能会超过您的 SNS 主题配额。也可以使用 CloudWatch 指标检查违规。有关更多信息，请参阅《AWS IoT Core 开发人员指南》中的使用 Amazon CloudWatch 监控 AWS IoT 警报和指标。
5. 使用 AttachSecurityProfile 操作将安全配置文件附加到一组设备（事物组）、账户中所有已注册的事物、所有未注册的事物或所有设备。AWS IoT Device DefenderDetect 开始检测异常行为，如果检测到任何行为违规，则发送告警。您可能想要将安全配置文件附加到所有未注册的事物，例如，您希望与不在账户事物注册表中的移动设备进行交互。您可以根据需求，为不同设备组定义不同的行为集。

   要将安全配置文件附加到一组设备，必须指定包含这些设备的事物组的 ARN。事物组 ARN 采用以下格式。

   ```
   arn:aws:iot:region:account-id:thinggroup/thing-group-name
   ```

要将安全配置文件附加到AWS 账户中所有已注册的事物（忽略未注册的事物），必须指定以下格式的 ARN：

```
arn:aws:iot:region:account-id:all/registered-things
```

要将安全配置文件附加到所有未注册的事物，必须指定以下格式的 ARN。

```
arn:aws:iot:region:account-id:all/unregistered-things
```

要将安全配置文件附加到所有设备，必须指定以下格式的 ARN。

```
arn:aws:iot:region:account-id:all/things
```

6. 也可以使用 ListActiveViolations 操作跟踪违规项，从而了解对于给定的安全配置文件或目标设备，检测到了哪些违规项。

   使用 ListViolationEvents 操作查看在指定时间段内检测到哪些违规项。您可以按安全配置文件、设备或告警验证状态筛选这些结果。
7. 通过使用 PutVerificationStateOnViolation 操作标记告警的验证状态并提供该验证状态的说明，可以验证、组织和管理报警。
8. 如果设备违反定义的行为过于频繁或太不频繁，那么应该调整行为定义。
9. 要查看您设置的安全配置文件和正在受到监控的设备，请使用 ListSecurityProfiles、ListSecurityProfilesForTarget 和 ListTargetsForSecurityProfile 操作。

   使用 DescribeSecurityProfile 操作获取有关安全配置文件的更多详细信息。
10. 要更新安全配置文件，请使用 UpdateSecurityProfile 操作。使用 DetachSecurityProfile 操作从账户或目标事物组分离安全配置文件。使用 DeleteSecurityProfile 操作完全删除安全配置文件。

缓解操作

您可以使用 AWS IoT Device Defender 以采取操作缓解 Audit 查找结果或 Detect 告警中发现的问题。

> **Note**
>
> 不会对被隐藏的审计查找结果执行缓解操作。有关审计查找结果隐藏的更多信息，请参阅 审计查找结果隐藏。

审计缓解操作

AWS IoT Device Defender 为不同的审计检查提供了预定义的操作。您可以为您的AWS 账户配置这些操作，然后将其应用于一组查找结果。这些结果可以是：

- 审核中发现的所有结果。此选项可通过 AWS IoT 控制台和 AWS CLI 来使用。
- 各个结果的列表。此选项仅在使用 AWS CLI 的情况下可用。
- 筛选的审核结果集。

下表列出了针对每种情况的审核检查类型和支持的缓解操作：

审核检查到缓解操作映射

审核检查	支持的缓解操作
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
UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS

审核检查	支持的缓解操作
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
IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK	PUBLISH_FINDING_TO_SNS

所有审计检查都支持向 Amazon SNS 发布审计查找结果，以便您可以采取自定义操作来响应通知。每种类型的审核检查可以支持其它缓解操作：

REVOKED_CA_CERT_CHECK

- 在 AWS IoT 中更改证书的状态以将其标记为非活动状态。

DEVICE_CERTIFICATE_SHARED_CHECK

- 在 AWS IoT 中更改设备证书的状态以将其标记为非活动状态。
- 将使用该证书的设备添加到事物组。

UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- 无额外受支持的操作。

AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- 无额外受支持的操作。

IOT_POLICY_OVERLY_PERMISSIVE_CHECK

- 添加一个空白 AWS IoT 策略版本以限制权限。

IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK

- 识别 AWS IoT 策略中的潜在错误配置。

CA_CERT_APPROACHING_EXPIRATION_CHECK

- 在 AWS IoT 中更改证书的状态以将其标记为非活动状态。

CONFLICTING_CLIENT_IDS_CHECK

- 无额外受支持的操作。

DEVICE_CERT_APPROACHING_EXPIRATION_CHECK

- 在 AWS IoT 中更改设备证书的状态以将其标记为非活动状态。
- 将使用该证书的设备添加到事物组。

DEVICE_CERTIFICATE_KEY_QUALITY_CHECK

- 在 AWS IoT 中更改设备证书的状态以将其标记为非活动状态。
- 将使用该证书的设备添加到事物组。

CA_CERTIFICATE_KEY_QUALITY_CHECK

- 在 AWS IoT 中更改证书的状态以将其标记为非活动状态。

REVOKED_DEVICE_CERT_CHECK

- 在 AWS IoT 中更改设备证书的状态以将其标记为非活动状态。
- 将使用该证书的设备添加到事物组。

LOGGING_DISABLED_CHECK

- 启用日志记录。

AWS IoT Device Defender 支持对 Audit 查找结果执行以下类型的缓解操作：

操作类型	注意事项
ADD_THINGS_TO_THING_GROUP	您指定要向其中添加设备的组。您还可以指定是否应覆盖一个或多个动态组中的成员身份（如果它超过了该事物可能所属组的最大数量）。
ENABLE_IOT_LOGGING	您可以指定日志记录级别和具有日志记录权限的角色。您不能指定 DISABLED 日志记录级别。
PUBLISH_FINDING_TO_SNS	您指定结果应发布到的主题。
REPLACE_DEFAULT_POLICY_VERSION	您指定模板名称。使用默认策略或空白策略替换策略版本。目前仅支持值 BLANK_POLICY 。
UPDATE_CA_CERTIFICATE	您指定 CA 证书的新状态。目前仅支持值 DEACTIVATE 。
UPDATE_DEVICE_CERTIFICATE	您指定设备证书的新状态。目前仅支持值 DEACTIVATE 。

通过配置在审核期间发现问题时要执行的标准操作，您可以一致地响应这些问题。使用这些定义的缓解操作还可以帮助您更快地解决问题，并减少人为错误的可能性。

> **Important**
>
> 应用旨在更改证书、将事物添加到新事物组或替换策略的缓解操作可能会对您的设备和应用程序产生影响。例如，设备可能无法连接。请考虑缓解操作的影响，然后再应用这些操作。您可能需要采取其它操作来纠正该问题，然后您的设备和应用程序才可正常工作。例如，您可能需要提供更新的设备证书。缓解操作可帮助您快速限制您的风险，但您仍必须采取纠正措施以解决底层问题。

某些操作（例如重新激活设备证书）只能手动执行。 AWS IoT Device Defender 不提供机制来自动回滚已应用的缓解操作。

检测缓解操作

AWS IoT Device Defender 支持对 Detect 告警执行以下类型的缓解操作：

操作类型	注意事项
ADD_THINGS_TO_THING_GROUP	您指定要向其中添加设备的组。您还可以指定是否应覆盖一个或多个动态组中的成员身份（如果它超过了该事物可能所属组的最大数量）。

如何定义和管理缓解操作

您可以使用 AWS IoT 控制台或 AWS CLI 为您的AWS 账户定义和管理缓解操作。

创建缓解操作

您定义的每个缓解操作是预定义的操作类型和特定于您账户的参数的组合。

使用 AWS IoT 控制台创建缓解操作

1. 打开 AWS IoT 控制台中的“Mitigation actions”（缓解操作）页面。
2. 在 Mitigation Actions（缓解操作）页面上，选择 Create（创建）。
3. 在 Create a new mitigation action（创建新的缓解操作）页面上，在 Action name（操作名称）中输入缓解操作的唯一名称。
4. 在操作类型中，指定您要定义的操作的类型。
5. 在 Permissions（权限）中，选择应用操作的权限所属的 IAM 角色。
6. 每个活动类型请求不同的一组参数。输入操作的参数。例如，如果选择将事物添加到事物组操作类型，请选择目标组，然后选择或清除 Override dynamic groups (覆盖动态组)。
7. 选择 Create（创建）以将缓解操作保存到您的 AWS 账户。

使用 AWS CLI 创建缓解操作

- 使用 CreateMitigationAction 命令创建您的缓解操作。当您将操作应用于审核结果时，将使用您为该操作提供的唯一名称。选择一个有意义的名称。

使用 AWS IoT 控制台查看和修改缓解操作

1. 打开 AWS IoT 控制台中的“Mitigation actions”（缓解操作）页面。

   Mitigation actions（缓解操作）页面显示为您的 AWS 账户定义的所有缓解操作的列表。
2. 选择您要更改的缓解操作的操作名称链接。
3. 选择 Edit（编辑），然后对缓解操作进行更改。由于缓解操作的名称用于标识此操作，因此，您不能更改此名称。
4. 选择 Update（更新）以将对缓解操作的更改保存到您的 AWS 账户。

使用 AWS CLI 列出缓解操作

- 使用 ListMitigationAction 命令可列出您的缓解操作。如果您想更改或删除缓解操作，请记下名称。

使用 AWS CLI 更新缓解操作

- 使用 UpdateMitigationAction 命令更改您的缓解操作。

使用 AWS IoT 控制台删除迁移操作

1. 打开 AWS IoT 控制台中的“Mitigation actions”（缓解操作）页面。

   Mitigation actions（缓解操作）页面显示为您的 AWS 账户定义的所有缓解操作。
2. 选择要删除的缓解操作，然后选择 Delete（删除）。
3. 在 Are you sure you want to delete（是否确定要删除）窗口中，选择 Delete（删除）。

使用 AWS CLI 删除缓解操作

- 使用 UpdateMitigationAction 命令更改您的缓解操作。

使用 AWS IoT 控制台查看缓解操作详细信息

1. 打开 AWS IoT 控制台中的“Mitigation actions”（缓解操作）页面。

   Mitigation actions（缓解操作）页面显示为您的 AWS 账户定义的所有缓解操作。
2. 选择您要查看的缓解操作的操作名称链接。

使用 AWS CLI 查看缓解操作详细信息

- 使用 DescribeMitigationAction 命令可查看您的缓解操作的详细信息。

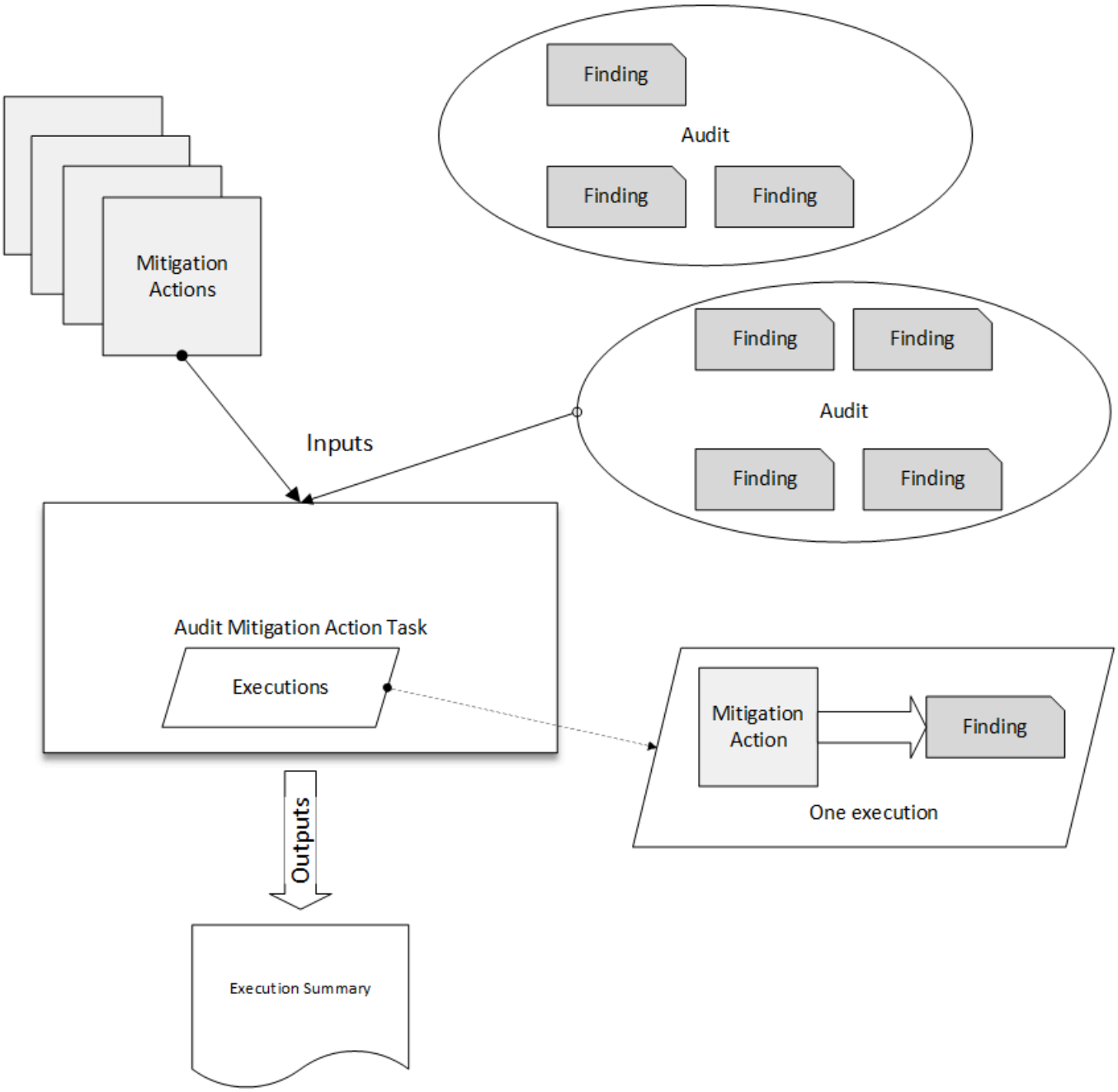
应用缓解操作

在定义了一组缓解操作后，您可以将这些操作应用于审核的结果。当您应用操作时，您就启动了审核缓解操作任务。此任务可能需要一些时间才能完成，具体取决于结果集和您应用于结果集的操作。例如，如果您有一个证书已过期的大型设备池，则可能需要一些时间来停用所有这些证书或将这些设备移到隔离组。其它操作（如启用日志记录）可以快速完成。

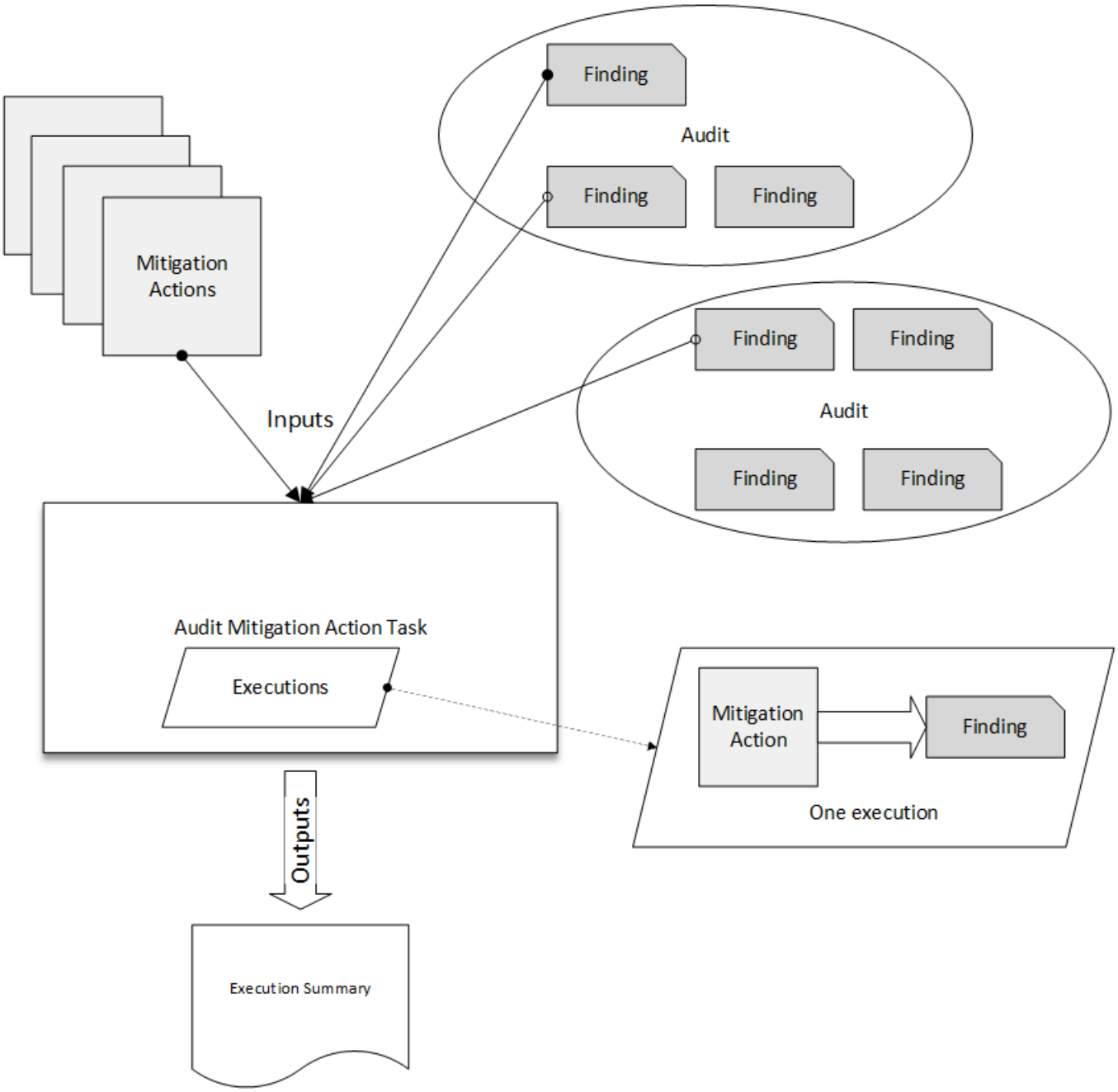
您可以查看操作执行的列表并取消尚未完成的执行。已作为已取消操作执行的一部分执行的操作不会回滚。如果您对一组结果应用多个操作而其中一个操作失败，则会对该结果跳过后续操作（但仍应用于其它结果）。结果的任务状态为 FAILED（失败）。如果在应用于结果时一个或多个操作失败，则 `taskStatus` 设置为失败。操作按指定的顺序应用。

每个操作执行将一组操作应用到目标。该目标可以是结果列表，也可以是来自某个审核的所有结果。

下图显示了您如何可以定义审核缓解任务，该任务从一个审核中获取所有结果并对这些结果应用一组操作。单次执行将一个操作应用于一个结果。审核缓解操作任务输出执行摘要。



下图显示了您如何可以定义审核缓解任务，该任务从一个或多个审核中获取各个结果的列表并对这些结果应用一组操作。单次执行将一个操作应用于一个结果。审核缓解操作任务输出执行摘要。



您可以使用 AWS IoT 控制台或 AWS CLI 以应用缓解操作。

使用 AWS IoT 控制台通过启动操作执行以应用缓解操作

1. 在 AWS IoT 控制台中打开“Audit results”（审计结果）页面。
2. 选择您要应用操作的审核的名称。
3. 选择 Start mitigation actions（启动缓解操作）。如果所有检查都符合要求，则此按钮不可用。

4. 在 Start a new mitigation action（启动新的缓解操作）中，任务名称默认为审核 ID，但您可以将其更改为更有意义的内容。
5. 对于在审核中具有一个或多个不合规结果的每种类型的检查，您可以选择一个或多个要应用的操作。仅显示对检查类型有效的操作。

> **Note**
>
> 如果尚未为AWS 账户配置操作，则操作列表为空。您可以选择 Create mitigation action（创建缓解操作）链接来创建一个或多个缓解操作。

6. 指定要应用的所有操作后，请选择 Start task（启动任务）。

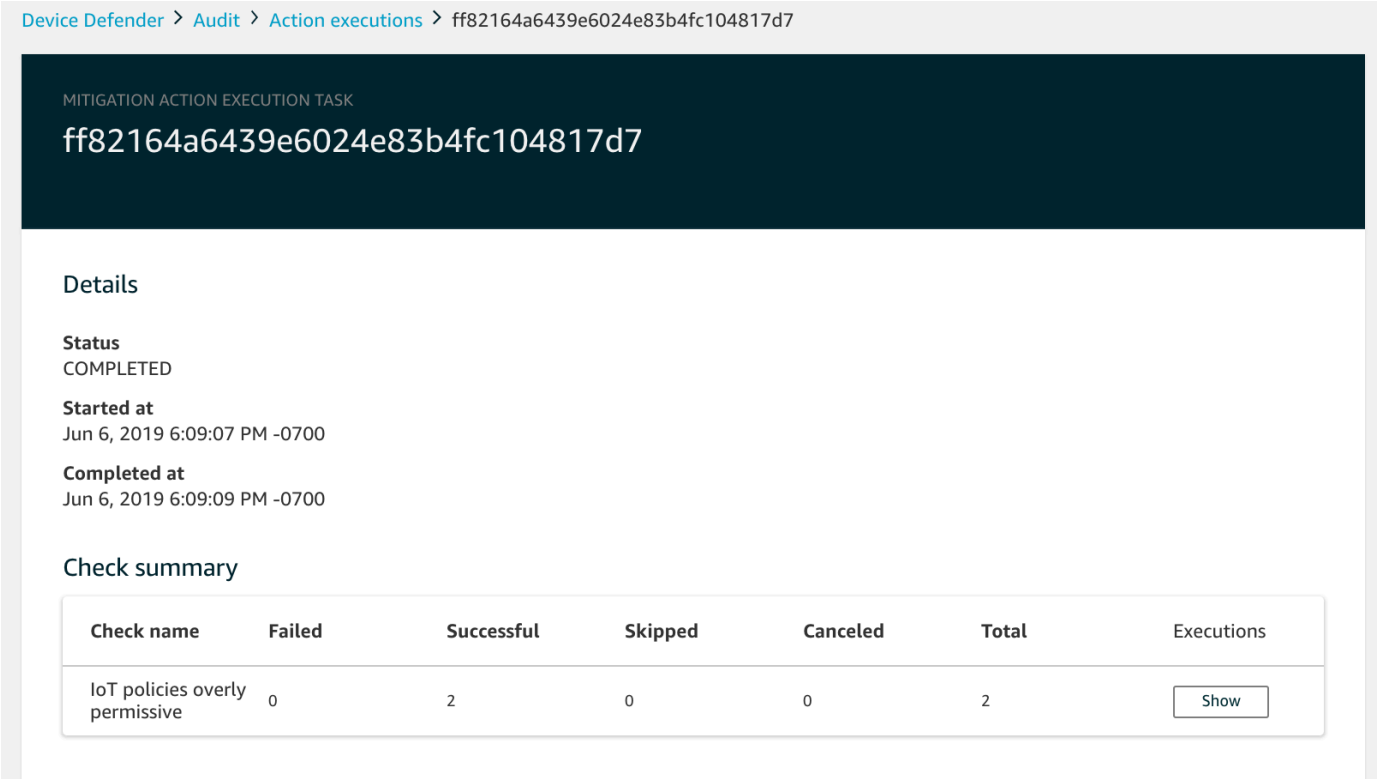
使用 AWS CLI 通过启动审核缓解操作执行以应用缓解操作

1. 如果您要对审计的所有查找结果应用操作，请使用 ListAuditTasks 命令查找任务 ID。
2. 如果您要仅对选定的查找结果应用操作，请使用 ListAuditFindings 命令获取结果 ID。
3. 使用 ListMitigationActions 命令，并记下要应用的缓解操作的名称。
4. 使用 StartAuditMitigationActionsTask 命令将操作应用到目标。记下任务 ID。您可以使用该 ID 来检查操作执行的状态，查看详细信息，或取消它。

使用 AWS IoT 控制台查看您的操作执行

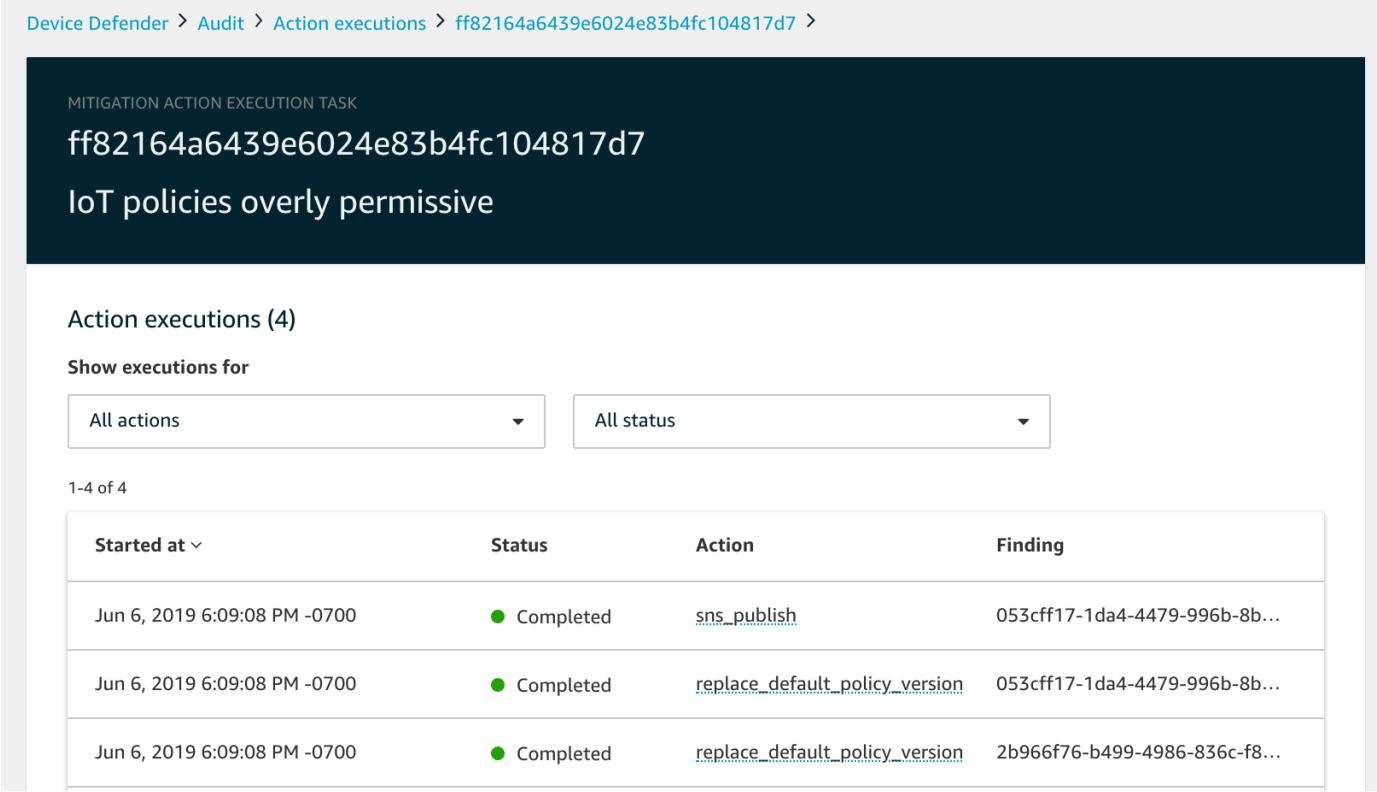
1. 打开 AWS IoT 控制台中的“Action tasks”（操作任务）页面。

   操作任务列表显示每个操作任务的启动时间和当前状态。
2. 选择名称链接以查看任务的详细信息。详细信息包括任务应用的所有操作、其目标及其状态。



您可以使用 Show executions for (显示的执行面向) 筛选条件来重点关注某些类型的操作或操作状态。

3. 要查看任务的详细信息，请在 Executions (执行) 中选择 Show (显示)。



使用 AWS CLI 列出已启动的任务

1. 使用 ListAuditMitigationActionsTasks 查看您的审计缓解操作任务。您可以提供筛选条件来缩小结果范围。如果要查看任务的详细信息，请记下任务 ID。
2. 使用 ListAuditMitigationActionsExecutions 查看特定审计缓解操作任务的执行详细信息。
3. 使用 DescribeAuditMitigationActionsTask 查看有关任务的详细信息，例如启动时指定的参数。

使用 AWS CLI 取消正在运行的审核缓解操作任务

1. 使用 ListAuditMitigationActionsTasks 命令找到您要取消其执行的任务的 ID。您可以提供筛选条件来缩小结果范围。
2. 使用 ListDetectMitigationActionsExecutions 命令通过任务 ID 取消您的审计缓解操作任务。您无法取消已完成的任务。当您取消任务时，不会应用剩余的操作，但不会回滚已应用的缓解操作。

权限

对于您定义的每个缓解操作，您必须提供用于应用该操作的角色。

缓解操作的权限

操作类型	权限策略模板

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"], "Resource":["*"]

操作类型	权限策略模板	
	<pre> }] }</pre>	

ADD_THINGS_TO_THING_GROUP	<pre>{
	    "Version":"2012-10-17",
	    "Statement":[
	        {
	            "Effect": "Allow",
	            "Action":[
	
	 "iot:ListPrincipalThings",
	
	 "iot:AddThingToThingGroup"
	            ],
	            "Resource": [
	                "*"
	            ]
	        }
	    ]
	}</pre>	

操作类型	权限策略模板	

REPLACE_DEFAULT_POLICY_VERSION	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"iot:CreatePolicyVersion"
	],
	"Resource":[
	"*"
	]
	}
	]
	}	

操作类型	权限策略模板	

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>*"
	]
	}
	]
	}	

操作类型	权限策略模板

PUBLISH_FINDING_TO_SNS	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect": "Allow",
	"Action":[
	"sns:Publish"
	],
	"Resource": [
	"*<The SNS topic to which the finding is published>* "
	]
	}
	]
	}

对于所有缓解操作类型，请使用以下信任策略模板：

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

缓解操作命令

您可以使用这些缓解操作命令为您的AWS 账户定义一组操作，以后可以将这些操作应用于一组或多组审计查找结果。有三种命令类别：

- 用于定义和管理操作的命令。
- 用于启动和管理将这些操作应用于 Audit 查找结果的命令。
- 用于启动和管理将这些操作应用 Detect 告警的命令。

缓解操作命令

定义和管理操作	启动和管理 Audit 执行	启动和管理 Detect 执行
CreateMitigationAction	CancelAuditMitigationActionsTask	CancelDetectMitigationActionsTask
DeleteMitigationAction	DescribeAuditMitigationActionsTask	DescribeDetectMitigationActionsTask
DescribeMitigationAction	ListAuditMitigationActionsTasks	ListDetectMitigationActionsTasks
ListMitigationActions	StartAuditMitigationActionsTask	StartDetectMitigationActionsTask
UpdateMitigationAction	ListAuditMitigationActionsExecutions	ListDetectMitigationActionsExecutions

将 AWS IoT Device Defender 与其它 AWS 产品结合使用

在运行 AWS IoT Greengrass 的设备上使用 AWS IoT Device Defender

AWS IoT Greengrass 提供了与 AWS IoT Device Defender 的预构建集成以持续监控设备行为。

- Device Defender 与 AWS IoT Greengrass V1 集成
- Device Defender 与 AWS IoT Greengrass V2 集成

将 AWS IoT Device Defender 与 FreeRTOS 和嵌入式设备搭配使用

要在 FreeRTOS 设备上使用 AWS IoT Device Defender，则您的设备必须安装了 FreeRTOS Embedded C SDK 或 AWS IoT Device Defender 库。FreeRTOS Embedded C SDK 包含 AWS IoT Device Defender 库。有关如何将 AWS IoT Device Defender 与您的 FreeRTOS 设备集成的信息，请参阅以下演示：

- AWS IoT Device Defender适用于 FreeRTOS 标准指标和自定义指标演示
- 使用 MQTT 代理将指标提交到 AWS IoT Device Defender
- 使用 MQTT 核心库将指标提交到 AWS IoT Device Defender

要在没有 FreeRTOS 的嵌入式设备上使用 AWS IoT Device Defender，您的设备就必须具有 AWS IoT 嵌入式 C 开发工具包或 AWS IoT Device Defender 库。AWS IoT Embedded C SDK 包括 AWS IoT Device Defender 库。有关如何集成 AWS IoT Device Defender 和嵌入式设备的信息，请参阅以下演示，AWS IoT Device Defender 了解 AWS IoT 嵌入式SDK标准和自定义指标演示。

配合使用 AWS IoT Device Defender和 AWS IoT Device Management

您可以使用 AWS IoT Device Management 实例集索引为索引、搜索和聚合您的 AWS IoT Device Defender 检测违规情况。在实例集索引中建立 Device Defender 违规数据之后，您可以访问和查询来自 Fleet Hub 应用程序的 Device Defender 违规数据，根据违规数据创建实例集告警以监控设备实例集中的异常情况，以及在 Fleet Hub 仪表板中查看实例集告警。

> **Note**
>
> 支持索引 AWS IoT Device Defender 违规数据的实例集索引特征已在 AWS IoT Device Management 的预览版本中，可能会发生更改。

- 管理实例集索引
- 查询语法
- 管理 Fleet Hub 应用程序的实例集索引
- 入门

与 AWS Security Hub 集成

AWS Security Hub 为您提供 AWS 安全状态的全面视图，可帮助您检查环境是否符合安全行业标准和最佳实践。Security Hub 从跨 AWS 账户、服务和受支持的第三方产品中收集安全数据。您可以使用 Security Hub 分析安全趋势并确定优先级最高的安全问题。

AWS IoT Device Defender 与 Security Hub 的集成使您能够从 AWS IoT Device Defender 中的 Security Hub 接收调查结果。Security Hub 会在其对您的安全状况分析中包含这些结果。

目录

- 启用和配置集成
- AWS IoT Device Defender 将结果发送到 Security Hub 的方式
  - AWS IoT Device Defender 发送的结果类型
  - 发送调查发现的延迟
  - 当 Security Hub 不可用时重试
  - 更新 Security Hub 中的现有 结果
- 来自 AWS IoT Device Defender 的典型结果
- 停止 AWS IoT Device Defender 向 Security Hub 发送调查结果

启用和配置集成

将 AWS IoT Device Defender 与 Security Hub 集成之前，您必须先启用 Security Hub。有关如何启用 Security Hub 的信息，请参阅 AWS Security Hub 用户指南中的设置 Security Hub。

同时启用 AWS IoT Device Defender 和 Security Hub 后，打开 Security Hub 控制台中的“Integrations”（集成）页面，然后选择 Accept findings for Audit, Detect, or both（接受审计或检测的调查结果或同时接受两者的调查结果）。AWS IoT Device Defender 将开始向 Security Hub 发送调查结果。

AWS IoT Device Defender 将结果发送到 Security Hub 的方式

在 Security Hub 中，安全问题按调查结果进行跟踪。一些调查结果来自其他 AWS 服务或第三方产品检测到的问题。

Security Hub 提供了管理来自所有这些来源的结果的工具。您可以查看和筛选结果列表，并查看结果的详细信息。有关更多信息，请参阅 AWS Security Hub 用户指南中的查看结果。您还可以跟踪调查发现的调查状态。有关更多信息，请参阅 AWS Security Hub 用户指南中对结果采取行动。

Security Hub 中的所有调查结果都使用名为 AWS 安全检测结果格式 (ASFF) 的标准 JSON 格式。ASFF 包含有关问题根源、受影响资源以及调查发现当前状态的详细信息。有关 ASFF 的更多信息，请参阅 AWS Security Hub 用户指南中的 AWS 安全检测结果格式 (ASFF)。

AWS IoT Device Defender 是一项 AWS 服务，可将检测结果发送到 Security Hub。

AWS IoT Device Defender 发送的结果类型

启用 Security Hub 集成后，AWS IoT Device Defender 审计会将其生成的检测结果（称为检查摘要）发送到 Security Hub。检查摘要是关于特定审计检查类型和特定审计任务的一般信息。有关更多信息，请参阅审计检查。

AWS IoT Device Defender 审计会向 Security Hub 发送有关每项审计任务中的“审计检查摘要”和“审计结果”的检测结果更新。如果在“审计检查”中找到的所有资源均符合要求，或者审计任务被取消，审计会将 Security Hub 中的检查摘要更新为“已存档”记录状态。如果某项资源在审计检查中被报告为不合规，但在上一次审计任务中被报告为合规，则审计会将其更改为合规，并将 Security Hub 中的调查结果更新为“已存档”记录状态。

AWS IoT Device Defender 检测会将违规调查结果发送到 Security Hub。这些违规调查结果包括机器学习 (ML)、统计和静态行为。

为了将调查结果发送到 Security Hub，AWS IoT Device Defender 使用 AWS 安全检测结果格式 (ASFF)。在 ASFF 中，`Types` 字段提供结果类型。来自 AWS IoT Device Defender 的结果可能具有 `Types` 的以下值。

不寻常的行为

冲突的 MQTT 客户端 ID 和设备证书共享检查的调查结果类型，以及检测的调查结果类型。

软件和配置检查/漏洞

所有其他审计检查的调查结果类型。

发送调查发现的延迟

AWS IoT Device Defender 审计创建新调查结果时，将在审计任务完成后立即将结果发送到 Security Hub。延迟取决于审计任务中生成的调查结果的数量。Security Hub 通常会在一小时内收到调查结果。

AWS IoT Device Defender 检测将近乎实时发送违规调查结果。在违规进入或退出警报（意味着已创建或删除警报）后，会立即创建或存档相应的 Security Hub 调查结果。

当 Security Hub 不可用时重试

如果 Security Hub 不可用，AWS IoT Device Defender 审计和 AWS IoT Device Defender 检测会重试发送调查结果，直到收到这些结果。

更新 Security Hub 中的现有 结果

将 AWS IoT Device Defender 审计调查结果发送到 Security Hub 后，您可以通过选中的资源标识符和审计检查类型对其进行识别。如果在后续审计任务中针对同一资源和审计检查生成了新的审计调查结果，AWS IoT Device Defender 审计会向 Security Hub 发送更新，以反映调查结果活动的其他观察结果。如果在后续审计任务中没有针对同一资源和审计检查生成其他审计调查结果，则资源将更改为符合审计检查。 AWS IoT Device Defender然后，审计会将调查结果存档到 Security Hub 中。

AWS IoT Device Defender 审计还会更新 Security Hub 中的检查摘要。如果在审计检查中发现不合规资源或检查失败，Security Hub 调查结果的状态将变为活动。否则，AWS IoT Device Defender 审计会将调查结果存档到 Security Hub 中。

当发生冲突（例如，在警报中）时，AWS IoT Device Defender 检测会创建一个 Security Hub 调查结果。仅当满足以下条件之一时，才会更新调查结果：

- 由于调查结果将很快在 Security Hub 中过期，因此 AWS IoT Device Defender 会发送更新以使该调查结果保持最新状态。调查结果将在最新更新后 90 天或创建日期后 90 天（如果未发生更新）被删除。有关更多信息，请参阅 AWS Security Hub 用户指南中的 Security Hub 配额。
- 相应的违规将解除警报，因此 AWS IoT Device Defender 会将其调查结果状态更新为“已存档”。

来自 AWS IoT Device Defender 的典型结果

AWS IoT Device Defender 使用 AWS 安全检测结果格式 (ASFF) 将调查结果发送到 Security Hub。

以下示例显示了 Security Hub 中审计调查结果的典型调查结果。`ProductFields` 中的 `ReportType` 是 `AuditFinding`。

```
  {
  "SchemaVersion": "2018-10-08",
  "Id": "336757784525/IOT_POLICY/policyexample/1/IOT_POLICY_OVERLY_PERMISSIVE_CHECK/
ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",
  "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/iot-device-defender-audit",
  "ProductName": "IoT Device Defender - Audit",
  "CompanyName": "AWS",
  "Region": "us-west-2",
  "GeneratorId": "1928b87ab338ee2f541f6fab8c41c4f5",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Check/Vulnerabilities"
  ],
  "CreatedAt": "2022-11-06T22:11:40.941Z",
  "UpdatedAt": "2022-11-06T22:11:40.941Z",
  "Severity": {
    "Label": "CRITICAL",
    "Normalized": 90
  },
  "Title": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK:
 ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",
  "Description": "IOT_POLICY policyexample:1 is reported as non-compliant for
 IOT_POLICY_OVERLY_PERMISSIVE_CHECK by Audit task 9f71b6e90cfb57d4ac671be3a4898e6a.
 The non-compliant reason is Policy allows broad access to IoT data plane actions:
 [iot:Connect].",
  "SourceUrl": "https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/
policy/policyexample",
  "ProductFields": {
    "CheckName": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",
    "TaskId": "9f71b6e90cfb57d4ac671be3a4898e6a",
    "TaskType": "ON_DEMAND_AUDIT_TASK",
    "PolicyName": "policyexample",
    "IsSuppressed": "false",
    "ReasonForNonComplianceCode": "ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",
    "ResourceType": "IOT_POLICY",
```

```
    "FindingId": "1928b87ab338ee2f541f6fab8c41c4f5",
    "PolicyVersionId": "1",
    "ReportType": "AuditFinding",
    "TaskStartTime": "1667772700554",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-west-2::product/
aws/iot-device-defender-audit/336757784525/IOT_POLICY/policyexample/1/
IOT_POLICY_OVERLY_PERMISSIVE_CHECK/ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",
    "aws/securityhub/ProductName": "IoT Device Defender - Audit",
    "aws/securityhub/CompanyName": "AWS"
  },
  "Resources": [
    {
      "Type": "AwsIotPolicy",
      "Id": "policyexample",
      "Partition": "aws",
      "Region": "us-west-2",
      "Details": {
        "Other": {
          "PolicyVersionId": "1"
        }
      }
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "CRITICAL"
    },
    "Types": [
      "Software and Configuration Check/Vulnerabilities"
    ]
  }
}
```

以下示例显示了 Security Hub 中审计检查摘要的典型调查结果。ProductFields 中的 ReportType 是 CheckSummary。

```
  {
  "SchemaVersion": "2018-10-08",
  "Id": "615243839755/SCHEDULED_AUDIT_TASK/daily_audit_schedule_checks/
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/iot-device-defender-audit",
  "ProductName": "IoT Device Defender - Audit",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "f3021945485adf92487c273558fcaa51",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Check/Vulnerabilities/CVE"
  ],
  "CreatedAt": "2022-10-18T14:20:13.933Z",
  "UpdatedAt": "2022-10-18T14:20:13.933Z",
  "Severity": {
    "Label": "CRITICAL",
    "Normalized": 90
  },
  "Title": "DEVICE_CERTIFICATE_KEY_QUALITY_CHECK Summary: Completed with 2 non-
compliant resources",
  "Description": "Task f3021945485adf92487c273558fcaa51 of weekly scheduled Audit
 daily_audit_schedule_checks completes. 2 non-cimpliant resources are found for
 DEVICE_CERTIFICATE_KEY_QUALITY_CHECK out of 1000 resources in the account. The
 percentage of non-compliant resources is 0.2%.",
  "SourceUrl": "https://us-east-1.console.aws.amazon.com/iot/home?region=us-east-1#/dd/
audit/results/f3021945485adf92487c273558fcaa51/DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
  "ProductFields": {
    "TaskId": "f3021945485adf92487c273558fcaa51",
    "TaskType": "SCHEDULED_AUDIT_TASK",
    "ScheduledAuditName": "daily_audit_schedule_checks",
    "CheckName": "DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
    "ReportType": "CheckSummary",
    "CheckRunStatus": "COMPLETED_NON_COMPLIANT",
    "NonComopliantResourcesCount": "2",
    "SuppressedNonCompliantResourcesCount": "1",
    "TotalResourcesCount": "1000",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/
iot-device-defender-audit/615243839755/SCHEDULED/daily_audit_schedule_checks/
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
    "aws/securityhub/ProductName": "IoT Device Defender - Audit",
    "aws/securityhub/CompanyName": "AWS"
  },
  "Resources": [
```

```
    {
      "Type": "AwsIotAuditTask",
      "Id": "f3021945485adf92487c273558fcaa51",
      "Region": "us-east-1"
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "CRITICAL"
    },
    "Types": [
      "Software and Configuration Check/Vulnerabilities/CVE"
    ]
  }
}
```

以下示例显示了 Security Hub 中 AWS IoT Device Defender 检测违规的典型调查结果。

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

停止 AWS IoT Device Defender 向 Security Hub 发送调查结果

要停止向 Security Hub 发送结果，您可以使用 Security Hub 控制台或 API。

有关更多信息，请参阅 AWS Security Hub 用户指南中的禁用和启用来自集成的结果流（控制台）或禁用来自集成的结果流（Security Hub API、AWS CLI）。

防止跨服务混淆座席

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在 AWS 中，跨服务模拟可能会导致混淆代理问题。一个服务（呼叫服务）调用另一项服务（所谓的服务）时，可能会发生跨服务模拟。可以操纵调用服务，通过调用服务使用其权限对另一个客户的资源进行操作，否则该服务不应有访问权限。为防止这种情况，AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

混淆代理安全问题会影响 AWS IoT Device Defender 从您处访问的三种资源：运行审计、发送安全配置文件违规事件的 SNS 通知以及运行缓解操作。对于其中每个操作，`aws:SourceArn` 的值必须如下所示：

- 对于在 UpdateAccountAuditConfiguration API 中传递的资源（RoleArn 和 notificationTarget RoleArn 属性），应将 `aws:SourceArn` 作为 `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:` 来缩小资源策略的范围。
- 对于在 CreateMitigationAction API 中传递的资源（RoleArn 属性），应将 `aws:SourceArn` 作为 `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:mitigationaction/`*`mitigationActionName`* 来缩小资源策略的范围。

- 对于在 CreateSecurityProfile API 中传递的资源（alertTargets 属性），应将 aws:SourceArn 作为 arn:*arnPartition*:iot:*region*:*accountId*:securityprofile/*securityprofileName* 来缩小资源策略的范围。

防范混淆代理问题最有效的方法是使用 aws:SourceArn 全局条件上下文键和资源的完整 ARN。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有通配符 (*) 的 aws:SourceArn 全局上下文条件键。例如 arn:aws:*servicename*:*:*123456789012*:*。

以下示例演示如何使用 AWS IoT Device Defender 中的 aws:SourceArn 和 aws:SourceAccount 全局条件上下文键来防范混淆代理问题。

```
 {
"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:"
    }
  }
}
}
```

设备代理的安全最佳实践

最低权限

应仅为代理进程授予履行其职责所需的最低权限。

基本机制

- 代理应该以非根用户身份运行。

- 代理应该以专用用户身份在其自己的组中运行。
- 应该为用户/组授予收集和传输指标所需资源的只读权限。
- 示例：针对代理示例的 /proc /sys 只读权限。
- 有关如何设置进程从而以较低权限运行的示例，请参阅 Python 示例代理附带的设置说明。

有一些知名的 Linux 机制可帮助您进一步限制或隔离代理进程：

高级机制

- CGroups
- SELinux
- Chroot
- Linux Namespaces

运营弹性

代理进程必须能够灵活应对意外的运营错误和异常，不能崩溃或永久退出。代码需要从容地处理异常，并且作为一项预防措施，它必须配置为在发生意外终止（例如，因为系统重启或未捕获的异常）时自动重启。

最少依赖项

代理在实施中必须使用尽可能少的依赖项（即第三方库）。如果出于任务复杂性（例如，传输层安全）这类正当理由使用了库，只能使用维护良好的依赖项，并建立机制以确保依赖项保持更新。如果添加的依赖项包含代理不使用的功能且默认处于活动状态（例如，打开的端口、域套接字），请在代码中禁用这些功能或通过库的配置文件禁用这些功能。

进程隔离

代理进程只能包含执行设备指标收集和传输所必需的功能。不得利用其它系统进程作为容器，或实施用于使用案例范围之外的功能。此外，代理进程必须避免创建入站通信渠道，如域套接字和网络服务端口，这种渠道将允许本地或远程进程干扰其操作并影响其完整性和隔离。

隐匿性

代理进程不得使用表明其用途和安全价值的关键字（如安全、监控或审核）命名。首选通用代码名称或每个设备唯一的随机进程名称。命名代理的二进制文件所在的目录以及进程参数的任何名称和值时，必须遵循相同的原则。

最少信息共享

部署到设备的任何代理项目均不得包含敏感信息，如特权凭证、调试代码和死码，或揭露有关代理收集类指标的服务器端处理详情或有关后端系统的其它相关详情的内联注释或文档文件。

传输层安全

要为数据传输建立 TLS 安全通道，代理进程必须在应用程序级别强制执行所有客户端验证（如果没有默认启用），例如证书链和域名验证。此外，代理必须使用包含受信任机构、不包含属于遭破坏证书发布者的证书的根证书存储区。

安全部署

任何代理部署机制（例如代码推送或同步）以及包含其二进制文件、源代码和任何配置文件（包括受信任的根证书）的存储库，都必须实施访问控制以防止未经授权的代码注入或篡改。如果部署机制依赖于网络通信，则使用加密方法来保护传输中的部署项目的完整性。

阅读更多内容

- AWS IoT Device Defender 中的安全性
- 了解 AWS IoT 安全模型
- Redhat：尝试 Python
- Python 中的 10 种常见安全问题和规避方法
- 什么是最小特权以及为什么需要它？
- 十大 OWASP 嵌入式安全实践
- OWASP IoT 项目

AWS IoT Device Defender 故障排除指南

帮助我们改进此主题

告诉我们如何优化内容

常规

问：使用 AWS IoT Device Defender 是否需要满足任何先决条件？

答：如果要使用设备报告的指标，必须首先在连接 AWS IoT 的设备或设备网关上部署代理。设备必须提供一致的客户端标识符或事物名称。

审核

问：我启用了检查，审核持续显示“In-Progress”已经很长时间了。是哪里出错了吗？ 何时可以看到结果？

答：检查一经启用，数据收集就会立即开始。不过，如果账户具有大量数据（例如证书、事物或策略）要收集，您可能要在启用检查后等候一段时间才能看到检查结果。

Detect

问：如何知道在 AWS IoT Device Defender 安全配置文件行为中设置的阈值？

答：首先创建低阈值的安全配置文件行为，并将其附加到包含相应设备组的事物组。您可以使用 AWS IoT Device Defender 查看当前指标，然后根据使用案例调整设备行为阈值。

问：我创建了一个行为，但它没有在预计的时间点触发违规。应该如何修复此问题？

答：在定义行为时，您指定了预期的设备的正常行为方式。例如，您有一个监控摄像头，只在 TCP 端口 8888 上连接一个中央服务器，您不希望它进行任何其它连接。要想在摄像头连接其它端口时收到提醒，您可以定义一个类似如下的行为：

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
```

```
    "value": {
      "ports": [ 8888 ]
    }
  }
}
```

如果摄像头在 TCP 端口 443 上建立 TCP 连接，则违反了设备行为并将触发提醒。

问：我的一个或多个行为违规。如何清除违规？

答：警报将在设备恢复预期行为后解除，如行为配置文件中所定义。系统在收到设备指标数据后即会评估行为配置文件。如果设备超过两天时间未发布任何指标，则违规事件将自动设置为 `alarm-invalidated`。

问：我删除了违规行为，但如何停止提醒？

答：删除行为会停止针对该行为的所有未来的违规和提醒。必须从通知机制中清除早期的提醒。行为被删除后，该行为的违规记录将同账户中的其它违规一样保留相同的时间。

设备指标

问：我在提交已知违反行为的指标报告，但没有触发任何违规。这是怎么回事？

答：请订阅以下 MQTT 主题以核对您的指标报告是否被接受：

```
$aws/things/THING_NAME/defender/metrics/FORMAT/rejected
$aws/things/THING_NAME/defender/metrics/FORMAT/accepted
```

其中，THING_NAME 是报告指标的事物的名称，FORMAT 是“JSON”或“CBOR ”，具体取决于事物提交的指标报告的格式。

订阅后，对于提交的每个指标报告，您应该会收到关于这些主题的消息。`rejected` 消息表明解析指标报告时出现了问题。消息负载中会包含一条错误消息，以帮助您更正指标报告中的任何错误。`accepted` 消息表明指标报告已正确解析。

问：如果在指标报告中发送空指标会怎样？

答：空的端口列表或 IP 地址列表始终被视为符合对应行为。如果对应行为违规，则违规将被清除。

问：为什么我的设备指标报告中包含不在 AWS IoT 注册表中的设备的消息？

如果您有一个或多个安全配置文件附加到所有事物或所有未注册的事物，那么 AWS IoT Device Defender 将包括未注册事物的指标。如果您要排除未注册事物的指标，可以将配置文件附加到所有已注册的设备，而不是所有设备。

问：即使我将安全配置文件附加到所有未注册的设备或所有设备，我也没有看到来自一个或多个未注册设备的消息。如何修复此问题？

请确认您正使用受支持的格式发送格式正确的指标报告。有关信息，请参阅 设备指标文档规范。验证未注册的设备是否使用的是一致的客户端标识符或事物名称。如果事物名称包含控制字符，或长度超过 128 字节的 UTF-8 编码字符，设备报告的消息将遭到拒绝。

问：如果将未注册的设备添加到注册表，或已注册的设备变为未注册，会发生什么情况？

答：如果在注册表中添加或删除设备：

- 如果它持续发布违规指标，您可以看到设备的两项单独的违规（一个在其注册的事物名称下，一个在其未注册的身份下）。旧身份的活动违规会在两天后不再显示，但在违规历史记录存在最长可达 14 天。

问：我应该在设备指标报告的报告 ID 字段提供什么值？

答：对每个指标报告使用唯一值，用正整数表示。常见的做法是使用 Unix 纪元时间戳。

问：我是否应该为 AWS IoT Device Defender 指标创建专用 MQTT 连接？

答：不需要使用单独的 MQTT 连接。

问：在连接以发布设备指标时，我应该使用什么客户端 ID？

对于 AWS IoT 注册表中的设备（事物），可使用注册的事物名称。对于不在 AWS IoT 注册表中的设备，在连接到 AWS IoT 时可使用一致的标识符。此做法有助于将违规与事物名称匹配。

问：我能否使用不同的客户端 ID 发布设备指标？

代表另一项事物发布指标是可以实现的。要实现此目的，您可以将指标发布到 AWS IoT Device Defender 为该设备预留的主题。例如，`Thing-1` 要发布自己的指标，并代表 `Thing-2` 发布指标。`Thing-1` 收集自己的指标并将其发布到 MQTT 主题：

```
$aws/things/Thing-1/defender/metrics/json
```

`Thing-1` 随后从 `Thing-2` 获取指标，并将这些指标发布到 MQTT 主题：

```
$aws/things/Thing-2/defender/metrics/json
```

问：我的账户中能拥有多少个安全配置文件和行为？

答：请参阅 AWS IoT Device Defender 终端节点和配额。

问：提醒目标的典型目标角色是什么样的？

答：允许 AWS IoT Device Defender 针对提醒目标（SNS 主题）发布提醒的角色需要具备以下两个条件：

- 信任关系，将 iot.amazonaws.com 指定为可信实体。
- 附加的策略，授权 AWS IoT 发布到指定的 SNS 主题。例如：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "<sns-topic-arn>"
        }
    ]
}
```

- 如果用于发布提示的 SNS 主题是加密主题，那么除了向 SNS 主题发布消息的权限外，AWS IoT 还必须再授予两个权限。例如：

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "sns:Publish",
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": "<sns-topic-arn>"
       }
    ]
}
```

问：带自定义指标类型 number 的指标报告提交失败，并显示错误消息 Malformed metrics report。这是怎么回事？

答：类型 number 只以单个指标值作为输入，但在 DeviceMetrics 报告中提交指标值时，必须将其作为具有单个值的数组传递。确保将指标值作为数组提交。

错误负载：

```
{"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}}}
```

错误消息：

```
{"thingName":"myThing","status":"REJECTED","statusDetails":
{"ErrorCode":"InvalidPayload","ErrorMessage":"Malformed metrics
 report"},"timestamp":1635802047699}
```

无错误有效负载：

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

响应：

```
{"thingName":"myThing","12334567":1635800375,"status":"ACCEPTED","timestamp":1635801636023}
```

AWS IoT Device Defender 中的安全性

AWS 的云安全性的优先级最高。为了满足对安全性最敏感的组织的需求，我们打造了具有超高安全性的数据中心和网络架构。作为 AWS 的客户，您也可以从这些数据中心和网络架构受益。

安全性是 AWS 和您的共同责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云的安全性 – AWS 负责保护在 AWS Cloud 中运行 AWS 服务的基础设施。AWS 还向您提供可安全使用的服务。第三方审核员定期测试和验证我们的安全性的有效性，作为 AWS Compliance Programs 的一部分。要了解适用于 AWS IoT Device Defender 的合规性计划，请参阅合规性计划范围内的 AWS 服务。
- 云中的安全性——您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

此文档将帮助您了解如何在使用 AWS IoT Device Defender 时应用责任共担模型。以下主题说明如何配置 AWS IoT Device Defender 以实现您的安全性和合规性目标。您还会了解如何使用其他 AWS 服务以帮助您监控和保护 AWS IoT Device Defender 资源。要详细了解 AWS IoT Core 中的安全性，请参阅《AWS IoT Core 开发人员指南》中的安全章节

主题

- AWS IoT Device Defender 中的数据保护
- 适用于 AWS IoT Device Defender 的身份和访问管理
- AWS IoT Device Defender 的合规性验证
- AWS IoT Device Defender 中的韧性

AWS IoT Device Defender 中的数据保护

AWS 责任共担模式适用于 AWS IoT Device Defender 中的数据保护。如该模式中所述，AWS 负责保护运行所有 AWS Cloud 的全球基础架构。您负责维护对托管在此基础架构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management（IAM）设置单个用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 使用 SSL/TLS 与 AWS 资源进行通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用 AWS CloudTrail 设置 API 和用户活动日记账记录。有关使用 CloudTrail 跟踪来捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的 Working with CloudTrail trails。
- 使用 AWS 加密解决方案以及 AWS 服务 中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果在通过命令行界面或 API 访问 AWS 时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅美国联邦信息处理标准（FIPS）140-3。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括通过控制台、API、AWS CLI 或 AWS SDK 使用 AWS IoT Device Defender 或其它 AWS 服务时。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

适用于 AWS IoT Device Defender 的身份和访问管理

AWS Identity and Access Management（IAM）是一项 AWS 服务，可以帮助管理员安全地控制对 AWS 资源的访问。IAM 管理员控制谁可以通过身份验证（登录）和获得授权（拥有权限）来使用 AWS IoT Device Defender 资源。IAM 是一项无需额外费用即可使用的 AWS 服务。

主题

- 受众
- 使用身份进行身份验证
- 使用策略管理访问
- AWS IoT Device Defender 如何与 IAM 协同工作
- 适用于 AWS IoT Device Defender 的基于身份的策略示例
- 对 AWS IoT Device Defender 身份和访问进行故障排除

受众

使用 AWS Identity and Access Management（IAM）的方式因您可以在 AWS IoT Device Defender 中执行的操作而异。

服务用户 – 如果使用 AWS IoT Device Defender 服务来完成任务，则管理员会为您提供所需的凭证和权限。当您使用更多 AWS IoT Device Defender 功能来完成工作时，您可能需要额外权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS IoT Device Defender 中的功能，请参阅对 AWS IoT Device Defender 身份和访问进行故障排除。

服务管理员 – 如果您在公司负责管理 AWS IoT Device Defender 资源，则您可能具有 AWS IoT Device Defender 的完全访问权限。您有责任确定您的服务用户应访问哪些 AWS IoT Device Defender 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要了解有关您的公司如何将 IAM 与 AWS IoT Device Defender 结合使用的更多信息，请参阅AWS IoT Device Defender 如何与 IAM 协同工作。

IAM 管理员 – 如果您是 IAM 管理员，您可能希望了解有关如何编写策略来管理对 AWS IoT Device Defender 的访问权限的详细信息。要查看您可在 IAM 中使用的 AWS IoT Device Defender 基于身份的策略示例，请参阅适用于 AWS IoT Device Defender 的基于身份的策略示例。

使用身份进行身份验证

身份验证是您使用身份凭证登录 AWS 的方法。您必须作为 AWS 账户根用户、IAM 用户或通过代入 IAM 角色进行身份验证（登录到 AWS）。

您可以使用通过身份源提供的凭证以联合身份登录到 AWS。AWS IAM Identity Center（IAM Identity Center）用户、您公司的单点登录身份验证以及您的 Google 或 Facebook 凭证都是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当您使用联合身份验证访问 AWS 时，您就是在间接代入角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录到 AWS 的更多信息，请参阅 AWS 登录 用户指南中的如何登录到您的 AWS 账户。

如果您以编程方式访问 AWS，则 AWS 将提供软件开发工具包 (SDK) 和命令行界面 (CLI)，以便使用您的凭证以加密方式签署您的请求。如果您不使用 AWS 工具，则必须自行对请求签名。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM User Guide 中的 AWS Signature Version 4 for API requests。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如，AWS 建议您使用多重身份验证（MFA）来提高账户的安全性。要了解更多信息，请参阅 AWS IAM Identity Center User Guide 中的 Multi-factor authentication和 IAM User Guide 中的 AWS Multi-factor authentication in IAM。

AWS 账户 根用户

当您创建 AWS 账户 时，最初使用的是一个对账户中所有 AWS 服务 和资源拥有完全访问权限的登录身份。此身份称为 AWS 账户 根用户，使用您创建账户时所用的电子邮件地址和密码登录，即可获得

该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅 IAM 用户指南中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）结合使用联合身份验证和身份提供程序，以使用临时凭证来访问 AWS 服务。

联合身份是来自企业用户目录、Web 身份提供程序、AWS Directory Service、Identity Center 目录的用户，或任何使用通过身份源提供的凭证来访问 AWS 服务 的用户。当联合身份访问 AWS 账户 时，他们代入角色，而角色提供临时凭证。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和组，也可以连接并同步到您自己的身份源中的一组用户和组以跨所有 AWS 账户 和应用程序使用。有关 IAM Identity Center 的信息，请参阅 AWS IAM Identity Center 用户指南中的什么是 IAM Identity Center？。

IAM 用户和群组

IAM 用户是 AWS 账户 内对某个人员或应用程序具有特定权限的一个身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅 IAM 用户指南中的对于需要长期凭证的使用场景定期轮换访问密钥。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅 IAM User Guide 中的 Use cases for IAM users。

IAM 角色

IAM 角色是 AWS 账户 中具有特定权限的身份。它类似于 IAM 用户，但与特定人员不关联。要在 AWS Management Console 中临时代入 IAM 角色，您可以从用户切换到 IAM 角色（控制台）。您可以调用 AWS CLI 或 AWS API 操作或使用自定义网址以担任角色。有关使用角色的方法的更多信息，请参阅 IAM 用户指南中的担任角色的方法。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关用于联合身份验证的角色的信息，请参阅《IAM 用户指南》中的针对第三方身份提供商创建角色（联合身份验证）。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅 AWS IAM Identity Center 用户指南中的 权限集。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅 IAM 用户指南中的 IAM 中的跨账户资源访问。
- 跨服务访问：某些 AWS 服务使用其它 AWS 服务中的特征。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Simple Storage Service (Amazon S3) 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话：当您使用 IAM 用户或角色在 AWS 中执行操作时，您将被视为主体。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用主体调用 AWS 服务 的权限，结合请求的 AWS 服务，向下游服务发出请求。只有在服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。
  - 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
  - 服务相关角色 – 服务相关角色是与 AWS 服务 关联的一种服务角色。服务可以代入代表您执行操作的角色。服务相关角色显示在您的 AWS 账户 中，并由该服务拥有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 Amazon EC2 上运行的应用程序 – 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时凭证。这优先于在 EC2 实例中存储访问密钥。要将 AWS 角色分配给 EC2 实例并使其对该实例的所有应用程序可用，您可以创建一个附加到实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM 角色向在 Amazon EC2 实例上运行的应用程序授予权限。

使用策略管理访问

您将创建策略并将其附加到 AWS 身份或资源，以控制 AWS 中的访问。策略是 AWS 中的对象；在与身份或资源相关联时，策略定义它们的权限。在主体（用户、根用户或角色会话）发出请求时，AWS

将评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略在 AWS 中存储为 JSON 文档。有关 JSON 策略文档的结构和内容的更多信息，请参阅 IAM 用户指南中的 JSON 策略概览。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。具有该策略的用户可以从 AWS Management Console、AWS CLI 或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅 IAM User Guide 中的 Define custom IAM permissions with customer managed policies。

基于身份的策略可以进一步归类为内联策略或托管策略。内联策略直接嵌入单个用户、组或角色中。托管式策略是可以附加到 AWS 账户 中的多个用户、组和角色的独立策略。托管式策略包括 AWS 托管式策略和客户管理型策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅 IAM User Guide 中的 Choose between managed policies and inline policies。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。主体可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用来自 IAM 的 AWS 托管策略。

访问控制列表 (ACL)

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Simple Storage Service (Amazon S3)、AWS WAF 和 Amazon VPC 是支持 ACL 的服务示例。要了解有关 ACL 的更多信息，请参阅 Amazon Simple Storage Service 开发人员指南中的访问控制列表（ACL）概览。

其他策略类型

AWS 支持额外的、不太常用的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界：权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 `Principal` 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅IAM 用户指南中的 IAM 实体的权限边界。
- 服务控制策略 (SCP) – SCP 是 JSON 策略，指定了组织或组织单元 (OU) 在 AWS Organizations 中的最大权限。AWS Organizations 服务可以分组和集中管理您的企业拥有的多个 AWS 账户 账户。如果在组织内启用了所有功能，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中实体（包括每个 AWS 账户根用户）的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的服务控制策略。
- 资源控制策略（RCP）：RCP 是 JSON 策略，您可以使用它们设置账户中资源的最大可用权限，而无需更新附加到您拥有的每个资源的 IAM 策略。RCP 限制了成员账户中资源的权限，并可能影响身份（包括 AWS 账户根用户）的有效权限，无论这些身份是否属于您的组织。有关 Organizations 和 RCP（包括支持 RCP 的 AWS 服务列表）的更多信息，请参阅《AWS Organizations 用户指南》中的 Resource control policies (RCPs)。
- 会话策略 – 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅 IAM 用户指南中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解 AWS 如何确定在涉及多种策略类型时是否允许请求，请参阅IAM 用户指南中的策略评估逻辑。

AWS IoT Device Defender 如何与 IAM 协同工作

在使用 IAM 管理对 AWS IoT Device Defender 的访问权限之前，请了解哪些 IAM 功能可以用于 AWS IoT Device Defender。

可以与 AWS IoT Device Defender 搭配使用的 IAM 功能

IAM 功能	AWS IoT Device Defender 支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键	是
ACL	否
ABAC（策略中的标签）	部分
临时凭证	是
主体权限	是
服务角色	是
服务相关角色	否

要大致了解 AWS IoT Device Defender 和其他 AWS 服务如何与大多数 IAM 功能一起使用，请参阅《IAM 用户指南》中的与 IAM 一起使用的 AWS 服务。

适用于 AWS IoT Device Defender 的基于身份的策略

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

适用于 AWS IoT Device Defender 的基于身份的策略示例

要查看 AWS IoT Device Defender 基于身份的策略的示例，请参阅适用于 AWS IoT Device Defender 的基于身份的策略示例。

AWS IoT Device Defender 内基于资源的策略

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。主体可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户存取，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当主体和资源处于不同的 AWS 账户 中时，则信任账户中的 IAM 管理员还必须授予主体实体（用户或角色）对资源的访问权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅IAM 用户指南中的 IAM 中的跨账户资源访问。

适用于 AWS IoT Device Defender 的策略操作

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

要查看 AWS IoT Device Defender 操作列表，请参阅《服务授权参考》。

AWS IoT Device Defender 中的策略操作在操作前使用以下前缀：

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
      ":action1",
      ":action2"
         ]
```

要查看 AWS IoT Device Defender 基于身份的策略的示例，请参阅适用于 AWS IoT Device Defender 的基于身份的策略示例。

AWS IoT Device Defender 的策略资源

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 `Resource` 或 `NotResource` 元素。作为最佳实践，请使用其 Amazon 资源名称 (ARN) 指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

要查看 AWS IoT Device Defender 资源类型及其 ARN 的列表，请参阅《服务授权参考》。要了解您可以使用哪些操作指定每个资源的 ARN，请参阅 。

要查看 AWS IoT Device Defender 基于身份的策略的示例，请参阅适用于 AWS IoT Device Defender 的基于身份的策略示例。

AWS IoT Device Defender 的策略条件键

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则 AWS 使用逻辑 OR 运算来评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅 IAM 用户指南中的 IAM 策略元素：变量和标签。

AWS 支持全局条件键和特定于服务的条件键。要查看所有 AWS 全局条件键，请参阅IAM 用户指南中的 AWS 全局条件上下文键。

有关 AWS IoT Device Defender 条件键的列表，请参阅《服务授权参考》。要了解您可以对哪些操作和资源使用条件键，请参阅 。

要查看 AWS IoT Device Defender 基于身份的策略的示例，请参阅适用于 AWS IoT Device Defender 的基于身份的策略示例。

AWS IoT Device Defender 中的 ACL

支持 ACL：否

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

带有 AWS IoT Device Defender 的 ABAC

支持 ABAC（策略中的标签）：部分支持

基于属性的访问控制 (ABAC) 是一种授权策略，该策略基于属性来定义权限。在 AWS 中，这些属性称为标签。您可以将标签附加到 IAM 实体（用户或角色）以及 AWS 资源。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 aws:ResourceTag/*key-name*、aws:RequestTag/*key-name* 或 aws:TagKeys 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅 IAM 用户指南中的使用基于属性的访问控制（ABAC）。

将临时凭证用于 AWS IoT Device Defender

支持临时凭证：是

某些 AWS 服务 在您使用临时凭证登录时无法正常工作。有关更多信息，包括 AWS 服务 与临时凭证配合使用，请参阅 IAM 用户指南中的使用 IAM 的 AWS 服务。

如果您不使用用户名和密码而用其它方法登录到 AWS Management Console，则使用临时凭证。例如，当您使用贵公司的单点登录（SSO）链接访问 AWS 时，该过程将自动创建临时凭证。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的从用户切换到 IAM 角色（控制台）。

您可以使用 AWS CLI 或者 AWS API 创建临时凭证。之后，您可以使用这些临时凭证访问 AWS。AWS 建议您动态生成临时凭证，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

AWS IoT Device Defender 的跨服务主体权限

支持转发访问会话（FAS）：是

当您使用 IAM 用户或角色在 AWS 中执行操作时，您将被视为主体。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用主体调用 AWS 服务 的权限，结合请求的 AWS 服务，向下游服务发出请求。只有在服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详细信息，请参阅转发访问会话。

AWS IoT Device Defender 的服务角色

支持服务角色：是

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

> **Warning**
>
> 更改服务角色的权限可能会破坏 AWS IoT Device Defender 的功能。只有在 AWS IoT Device Defender 提供指导时，才能编辑服务角色。

AWS IoT Device Defender 的服务相关角色

支持服务相关角色：否

服务相关角色是一种与 AWS 服务 相关的服务角色。服务可以代入代表您执行操作的角色。服务相关角色显示在您的 AWS 账户 中，并由该服务拥有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅能够与 IAM 搭配使用的 AWS 服务。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

适用于 AWS IoT Device Defender 的基于身份的策略示例

默认情况下，用户和角色没有创建或修改 AWS IoT Device Defender 资源的权限。他们也无法使用 AWS Management Console、AWS Command Line Interface（AWS CLI）或 AWS API 执行任务。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关 AWS IoT Device Defender 定义的操作和资源类型的详细信息，包括每种资源类型的 ARN 格式，请参阅《服务授权参考》中的 Actions, Resources, and Condition Keys for AWS IoT Device Defender。

主题

- 策略最佳实践
- 使用 AWS IoT Device Defender 控制台
- 允许用户查看他们自己的权限

策略最佳实践

基于身份的策略确定某个用户是否可以创建、访问或删除您账户中的 AWS IoT Device Defender 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- AWS 托管策略及转向最低权限许可入门 – 要开始向用户和工作负载授予权限，请使用 AWS 托管策略来为许多常见使用场景授予权限。您可以在 AWS 账户 中找到这些策略。我们建议通过定义特定于您的使用场景的 AWS 客户管理型策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管策略或工作职能的 AWS 托管策略。

- 应用最低权限 – 在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果通过特定（AWS 服务例如 AWS CloudFormation）使用服务操作，您还可以使用条件来授予对服务操作的访问权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅 IAM User Guide 中的 Validate policies with IAM Access Analyzer。
- 需要多重身份验证（MFA）– 如果您所处的场景要求您的 AWS 账户 中有 IAM 用户或根用户，请启用 MFA 来提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅 IAM User Guide 中的 Secure API access with MFA。

有关 IAM 中的最佳实操的更多信息，请参阅 IAM 用户指南中的 IAM 中的安全最佳实操。

使用 AWS IoT Device Defender 控制台

要访问 AWS IoT Device Defender 控制台，您必须拥有一组最低的权限。这些权限必须允许您列出和查看有关您的 AWS 账户中的 AWS IoT Device Defender 资源的详细信息。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于只需要调用 AWS CLI 或 AWS API 的用户，无需为其提供最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色仍可使用 AWS IoT Device Defender 控制台，请同时将 AWS IoT Device Defender *ConsoleAccess* 或 *ReadOnly* AWS 托管式策略添加到实体。有关更多信息，请参阅《IAM 用户指南》中的为用户添加权限。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管策略。此策略包括在控制台上完成此操作或者以编程方式使用 AWS CLI 或 AWS API 所需的权限。

```
{
    "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": "*"
        }
    ]
}
```

对 AWS IoT Device Defender 身份和访问进行故障排除

使用以下信息可协助您诊断和修复在使用 AWS IoT Device Defender 和 IAM 时可能遇到的常见问题。

主题

- 我无权在 AWS IoT Device Defender 中执行操作
- 我无权执行 iam:PassRole
- 我希望允许我的 AWS 账户以外的人访问我的 AWS IoT Device Defender 资源

我无权在 AWS IoT Device Defender 中执行操作

如果您收到错误提示，表明您无权执行某个操作，则您必须更新策略以允许执行该操作。

当 `mateojackson` IAM 用户尝试使用控制台查看有关虚构 *`my-example-widget`* 资源的详细信息，但不拥有虚构 *`:GetWidget`* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to
 perform: :GetWidget on resource: my-example-widget
```

在此情况下，必须更新 `mateojackson` 用户的策略，以允许使用 *`:GetWidget`* 操作访问 *`my-example-widget`* 资源。

如果您需要帮助，请联系 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam:PassRole

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 AWS IoT Device Defender。

有些 AWS 服务 允许将现有角色传递到该服务，而不是创建新服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 `marymajor` 的 IAM 用户尝试使用控制台在 AWS IoT Device Defender 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系 AWS 管理员。您的管理员是提供登录凭证的人。

我希望允许我的 AWS 账户以外的人访问我的 AWS IoT Device Defender 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解 AWS IoT Device Defender 是否支持这些功能，请参阅AWS IoT Device Defender 如何与 IAM 协同工作。
- 要了解如何为您拥有的 AWS 账户中的资源提供访问权限，请参阅《IAM 用户指南》中的为您拥有的另一个 AWS 账户中的 IAM 用户提供访问权限。
- 要了解如何为第三方 AWS 账户 提供您的资源的访问权限，请参阅 IAM 用户指南中的为第三方拥有的 AWS 账户 提供访问权限。
- 要了解如何通过联合身份验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（联合身份验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

AWS IoT Device Defender 的合规性验证

要了解某个 AWS 服务是否在特定合规性计划范围内，请参阅合规性计划范围内的 AWS 服务，然后选择您感兴趣的合规性计划。有关常规信息，请参阅 AWS 合规性计划、、。

您可以使用 AWS Artifact 下载第三方审计报告。有关更多信息，请参阅在 AWS Artifact 中下载报告、。

您使用 AWS 服务的合规性责任取决于您数据的敏感度、贵公司的合规性目标以及适用的法律法规。AWS 提供以下资源来帮助满足合规性：

- Security Compliance & Governance：这些解决方案实施指南讨论了架构考虑因素，并提供了部署安全性和合规性功能的步骤。
- 符合 HIPAA 要求的服务参考：列出符合 HIPAA 要求的服务。并非所有 AWS 服务 都符合 HIPAA 要求。
- AWS 合规性资源：此业务手册和指南集合可能适用于您的行业和位置。
- AWS 客户合规指南：从合规角度了解责任共担模式。这些指南总结了保护 AWS 服务 的最佳实践，并将指南映射到跨多个框架的安全控制，包括美国国家标准与技术研究院（NIST）、支付卡行业安全标准委员会（PCI）和国际标准化组织（ISO）。
- AWS Config 开发人员指南中的使用规则评估资源 - 此 AWS Config 服务评测您的资源配置对内部实践、行业指南和法规的遵循情况。
- AWS Security Hub：此 AWS 服务 向您提供 AWS 中安全状态的全面视图。Security Hub 通过安全控制措施评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控制措施的列表，请参阅 Security Hub 控制措施参考。

- Amazon GuardDuty：该 AWS 服务 通过监控您的环境中是否存在可疑和恶意活动，来检测您的 AWS 账户、工作负载、容器和数据面临的潜在威胁。GuardDuty 可以通过满足某些合规性框架规定的入侵检测要求，来协助您满足各种合规性要求，如 PCI DSS。
- AWS Audit Manager：此 AWS 服务 可帮助您持续审核您的 AWS 使用情况，以简化管理风险以及与相关法规和行业标准的合规性的方式。

AWS IoT Device Defender 中的韧性

AWS 全球基础设施围绕 AWS 区域和可用区构建。AWS 区域提供多个在物理上独立且隔离的可用区，这些可用区与延迟率低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础架构相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅 AWS 全球基础设施。

除了 AWS 全球基础设施之外，AWS IoT Device Defender 还提供了多种功能，以协助支持您的数据韧性和备份需求。

AWS IoT Device Defender 用户指南的文档历史记录

下表介绍了 AWS IoT Device Defender 的文档版本。

变更	说明	日期
正式发布	这是 AWS IoT Device Defender 的最初公开发布版。	2023 年 8 月 2 日
AWS IoT Device Defender 现在支持监控设备断开连接持续时间	AWS IoT Device Defender Rules Detect 现在支持新的断开连接持续时间指标，用于监控每台设备的断开连接持续时间。借助此附加指标，您可以跟踪设备断开连接的时间长短，以便了解设备是否按预期运行。您还可以在预定义的阈值级别上配置警报，并在设备持续出现连接问题时收到警报。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2023 年 7 月 20 日
AWS IoT Device Defender 审计功能可识别 IoT 策略中可能存在的配置错误	使用审计功能识别缺陷、排查问题并采取必要的纠正措施。这项新功能还有助于识别具有宽松允许语句的 IoT 策略，这种策略会让设备有可能访问意外的资源。它还检查拒绝语句中使用 MQTT 通配符的情况，而在此类情况下，当使用特定字符串替换通配符时，设备可能会绕过拒绝限制。有关更多信息，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标	2022 年 12 月 6 日

AWS IoT Device Defender ML Detect 自定义指标和维度支持	AWS IoT Device Defender 现在支持对已撤销的中间证书颁发机构（CA）进行新的审计检查。如果某个 CA 因为中间 CA 可能遭盗用而撤销该中间 CA，则该中间 CA 颁发的所有证书也可能遭盗用而无效。这项新的审计检查可识别已撤销的中间 CA 颁发的活动设备证书，并有助于客户查看和替换这些活动的设备证书。有关更多信息，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标	2022 年 11 月 10 日
AWS IoT Device Defender ML Detect 自定义指标和维度支持	ML Detect 现在支持监控自定义指标，使您能够评估您的队列特有的运行状况指标。除了使用 Rules Detect 手动设置静态警报外，您现在还可以使用机器学习来自动了解队列对自定义指标的预期行为。此外，借助 ML Detect 对新维度筛选条件的支持，您可以在机器学习安全配置文件中定义属性来评估更精确的指标。《AWS IoT Device Defender 开发人员指南》中的云端指标	2022 年 9 月 14 日

AWS IoT Device Management 和 AWS IoT Device Defender 现在支持通过 ListMetricValues API 监控设备指标	使用 ListMetricValues API，可以访问属于某个安全配置文件的联网设备的历史设备端、云端和自定义指标。除了在 AWS IoT 管理控制台中查看数据外，您现在还可以灵活地以编程方式监控和构建自己的可视化数据。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2022 年 4 月 5 日
AWS IoT Device Defender 现在支持 Detect 警报验证状态	根据对检测到的行为异常的调查结果来验证警报。这些调查结果可以验证警报到底是真实警报、良性警报、误报或未知，并提供验证说明。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2021 年 9 月 24 日
AWS IoT Device Defender 一键式审计功能发布	一键式审计功能使 AWS IoT Core 客户只需单击一下，即可根据安全最佳实践开始审计其账户和 IoT 设备，从而轻松改善其安全基准。一键式审计功能可让客户使用预设配置开启 AWS IoT Device Defender 审计，包括启用所有可用的审计检查和每日审计计划。它还提供有关定期安全审计的优势的上下文解释。一键式审计功能仅可从 AWS IoT 控制台获得。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2021 年 9 月 22 日

AWS IoT Device Defender CloudFormation 支持	AWS IoT Device Defender Rules Detect 现在支持新的断开连接持续时间指标来监控断开连接的持续时间。AWS IoT Device Defender 现在支持 AWS CloudFormation 以安全、高效和可重复的方式创建和配置 AWS IoT Device Defender 资源，例如计划的审计和安全配置文件。要详细了解 AWS IoT Device Defender 支持的 AWS CloudFormation 资源类型，请访问 IoT 资源类型参考。	2021 年 3 月 5 日
AWS IoT Device Defender 增加了对自定义指标的支持	使用 AWS IoT Device Defender 监控您的队列或使用案例所特有的运行状况指标。警报可以在 Device Defender 控制台中查看，也可以通过 AWS Simple Notification Service（SNS）进行共享。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2020 年 12 月 15 日
AWS IoT Device Defender 推出审计查找结果隐藏功能	审计查找结果隐藏功能可让您选择要查看的审计查找结果，并关闭特定资源的不合规查找结果。此外，您还可以在定义的时间段内或无限期地配置审计查找结果隐藏。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的审计。	2020 年 8 月 12 日

AWS IoT Device Defender 现在支持用于基于主题的指标监控的维度	维度功能使客户能够按照 MQTT 主题来筛选 Device Defender Detect 评估的指标。维度支持以下云端指标：收到的消息数量、消息字节大小、发送的消息数量、源 IP 和授权失败次数。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2020 年 4 月 2 日
AWS IoT Device Defender ML Detect 正式发布	AWS IoT Device Defender 的 ML Detect 通过从过往数据中进行学习，自动检测队列中的设备级操作和安全异常。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2020 年 3 月 24 日
AWS IoT Device Defender 为其审计功能添加了四项新检查	使用 AWS IoT Device Defender 审计功能来检查您的队列中是否有符合以下情况的设备：具有过度宽松的权限、可以访问超过 365 天未使用的服务、在基于 Debian 的操作系统上使用 OpenSSL 版本（已被确定为具有可预测的加密密钥，使其易受到暴力攻击），或者使用 Infineon RSA 库版本（已被确定为 RSA 密钥生成处理不当，使其易受到黑客攻击）。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的审计。	2019 年 11 月 25 日

AWS IoT Device Defender 对于审计结果支持缓解操作	AWS IoT Device Defender 支持客户将缓解操作应用于审计查找结果的功能。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的审计。	2019 年 8 月 6 日
AWS IoT Device Defender 支持监控未注册设备的行为	识别未在 AWS IoT Core 注册表中注册的设备的异常行为。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2019 年 5 月 15 日
AWS IoT Device Defender 现在提供统计异常检测和数据可视化	使用统计异常检测，并在设备不在基于百分比的阈值内时接收警报。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2019 年 2 月 19 日
AWS IoT Device Defender 现在支持监控设备断开连接持续时间	AWS IoT Device Defender 现在支持另外两个云端指标，即连接尝试次数和断开连接次数。有关文档，请参阅《AWS IoT Device Defender 开发人员指南》中的云端指标。	2018 年 12 月 19 日