Fügen Sie Eigenschaften der Geräte-Cloud-Instanz über die REST-API hinzu

Veröffentlicht: 2024-09-26

Mithilfe der Cloud-Eigenschaften von Geräten können Sie Informationen über Ihre Cloud-Umgebung im ExtraHop-System einsehen. Sie können den Namen, den Typ und die ID der Cloud-Instanz eines Geräts zusammen mit dem Cloud-Konto, dem das Gerät gehört, und der ID der Virtual Private Cloud , in der sich das Gerät befindet, identifizieren.

Dieses Handbuch enthält Anweisungen zum Hinzufügen einer Beobachtung über den ExtraHop API Explorer, eine AWS CloudFormation-Vorlage, eine AWS Lambda-Funktion und ein Python-Skript für Microsoft Azure. Wenn Sie Cloud-Eigenschaften automatisch über die REST-API aktualisieren, können Sie kontinuierlich Informationen von Ihrem Cloud-Anbieter abrufen, um sicherzustellen, dass Ihre Cloud-Eigenschaftsinformationen immer auf dem neuesten Stand sind.

Fügen Sie Cloud-Instanz-Eigenschaften über den ExtraHop API Explorer hinzu

Bevor Sie beginnen

- Für Sensoren und ECA-VMs benötigen Sie einen gültigen API-Schlüssel mit vollem Schreibzugriff Privilegien oder höher. (siehe Generieren Sie einen API-Schlüssel.)
- Für RevealX 360 benötigen Sie gültige REST-API-Anmeldeinformationen mit vollem Schreibzugriff Privilegien oder höher. (siehe REST-API-Anmeldeinformationen erstellen.)

1. Navigieren Sie in einem Browser zum REST API Explorer.

   Die URL ist der Hostname oder die IP-Adresse Ihres Sensor oder Konsole, gefolgt von `/api/v1/explore/`. Wenn Ihr Hostname beispielsweise seattle-eda ist, lautet die URL `https://seattle-eda/api/v1/explore/`.
2. Geben Sie Ihre REST-API-Anmeldeinformationen Anmeldedaten.
   - Für Sensoren und ECA-VMs klicken Sie auf **API-Schlüssel eingeben** und fügen Sie dann Ihren API-Schlüssel ein oder geben Sie ihn in das **API-Schlüssel** Feld.
   - Klicken Sie für RevealX 360 auf **Geben Sie die API-Anmeldeinformationen ein** und fügen Sie dann die ID und das Geheimnis Ihrer API-Anmeldeinformationen ein oder geben Sie sie in das **ID** und **Geheim** Felder.
3. Klicken Sie **Autorisieren** und klicken Sie dann **Schliessen**.
4. Finden Sie die ID des Gerät, indem Sie nach der MAC-Adresse des Gerät suchen.
   a) Klicken Sie **Gerät** und klicken Sie dann **POST /Geräte/Suche**.
   b) Klicken Sie **Probiere es aus**.
   c) Geben Sie im Textfeld den folgenden JSON-Code an und ersetzen Sie MACADDRESS durch die MAC-Adresse Ihres Cloud-Geräts:

   ```
   {
     "filter": {
       "field": "macaddr",
       "operand": "MACADDRESS",
       "operator": "="
     }
   }
   ```

   d) Klicken Sie **Anfrage senden**.
   e) Sehen Sie sich im Abschnitt Antworttext den Wert von an und Datensatz Sie ihn auf `id` Feld für jedes Gerät, das zurückgegeben wird.
5. Fügen Sie die Metadaten des Cloud-Geräts hinzu.

a) Klicken Sie **PATCH /geräte/ {id}**.
b) Klicken Sie **Probiere es aus**.
c) In der `id` Feld, geben Sie eine ID an.
d) Geben Sie im Textfeld den folgenden JSON-Code an und ersetzen Sie den `string` Werte mit Eigenschaften aus Ihrer Cloud-Umgebung:

```
{
  "cloud_account": "string",
  "cloud_instance_id": "string",
  "cloud_instance_name": "string",
  "cloud_instance_type": "string",
  "vpc_id": "string"
}
```

e) Klicken Sie **Anfrage senden**.

Fügen Sie AWS-Eigenschaften zu RevealX 360 mit CloudFormation hinzu

Sie können RevealX 360 mit einer CloudFormation-Vorlage, die auf Amazon S3 öffentlich verfügbar ist, Eigenschaften der AWS-Geräte-Cloud-Instanz hinzufügen. Die CloudFormation-Vorlage erstellt eine Lambda-Funktion, die Eigenschaften der AWS EC2-Instanz abruft und sie über die REST-API an RevealX 360 sendet. Die Lambda-Funktion ordnet Netzwerkschnittstellen von EC2-Instances Geräten zu, die auf dem ExtraHop-System anhand der MAC-Adresse erkannt wurden.

Hier sind einige wichtige Überlegungen zur Lambda-Funktion:

- Der AWS EventBridge-Service führt die Lambda-Funktion alle 30 Minuten aus.
- Die Funktion importiert nur Cloud-Instance-Eigenschaften für EC2-Instances.
- Sie müssen die CloudFormation-Vorlage in jedem AWS-Konto bereitstellen, aus dem Sie Eigenschaften importieren möchten.
- Sie können die Funktion nur in den folgenden AWS-Regionen bereitstellen:
  - USA Ost (Ohio)
  - USA Ost (Nord-Virginia)
  - USA West (Oregon)
  - USA West (Nordkalifornien)

  Informationen zum Hinzufügen von AWS-Eigenschaften außerhalb dieser Regionen finden Sie unter Fügen Sie AWS-Eigenschaften zu RevealX Enterprise mit Lambda hinzu.
- RevealX Enterprise unterstützt die CloudFormation-Vorlage nicht. Informationen zum Importieren von Eigenschaften in RevealX Enterprise finden Sie unter Fügen Sie AWS-Eigenschaften zu RevealX Enterprise mit Lambda hinzu.

Bevor Sie beginnen
Das musst du haben gültige REST-API-Anmeldeinformationen mit vollständigem Schreiben Privilegien oder höher.

1. Navigieren Sie zur CloudFormation-Seite in AWS.
2. Erstellen Sie einen CloudFormation-Stack aus der folgenden Amazon S3-URL:

   ```
   https://s3.us-east-2.amazonaws.com/ct.s.extrahoplabs/Public/MDS.yml
   ```

3. Konfigurieren Sie die folgenden Variablen:

   API-ID

   Die ID Ihrer RevealX 360 REST-API-Anmeldeinformationen.

   API-Geheimnis

   Das Geheimnis Ihrer RevealX 360 REST-API-Anmeldeinformationen.

Name des Mieters

Die Subdomain Ihrer RevealX 360-Konsole.

Weitere Informationen zur Konfiguration eines CloudFormation-Stacks finden Sie in der AWS-Dokumentation.

Fügen Sie AWS-Eigenschaften zu RevealX Enterprise mit Lambda hinzu

Sie können RevealX Enterprise mit einem Python-Beispielskript die Eigenschaften der AWS-Geräte-Cloud-Instanz hinzufügen. Das Skript ordnet Netzwerkschnittstellen von EC2-Instances Geräten zu, die auf dem ExtraHop-System anhand der MAC-Adresse erkannt wurden.

Hinweis: Informationen zum Importieren von AWS-Eigenschaften in RevealX 360 finden Sie unter Fügen Sie AWS-Eigenschaften zu RevealX 360 mit CloudFormation hinzu.

Das Skript ist so konzipiert, dass es als Lambda-Funktion in AWS ausgeführt wird. Hier sind einige wichtige Überlegungen zur Ausführung des Skripts in AWS:

- Das Skript ist so konzipiert, dass es in einem festgelegten Zeitintervall ausgeführt wird. Jedes Mal, wenn das Skript ausgeführt wird, scannt es jede Instanz auf der VPC und aktualisiert die entsprechenden Geräte im ExtraHop-System. Informationen zur Konfiguration einer Lambda-Funktion für die regelmäßige Ausführung finden Sie im AWS-Tutorial hier.
- Die Lambda-Funktion muss auf Ressourcen auf Ihrer VPC zugreifen können. Weitere Informationen finden Sie im AWS-Tutorial hier.
- Die Lambda-Funktion muss Listen- und Lesezugriff auf die DescribeInstances-Aktion für den EC2-Dienst haben. Weitere Informationen finden Sie im AWS-Tutorial hier.

Hinweis: Wenn das Skript eine Fehlermeldung zurückgibt, dass die TLS-Zertifikatsüberprüfung fehlgeschlagen ist, stellen Sie sicher, dass Ihrem Sensor oder Ihrer Konsole wurde ein vertrauenswürdiges Zertifikat hinzugefügt. Alternativ können Sie das hinzufügen `verify=False` Option zur Umgehung der Zertifikatsüberprüfung. Diese Methode ist jedoch nicht sicher und wird nicht empfohlen. Der folgende Code sendet eine HTTP GET-Anfrage ohne Zertifikatsüberprüfung:

```
requests.get(url, headers=headers, verify=False)
```

Bevor Sie beginnen

- Du musst eine haben gültiger API-Schlüssel mit vollständigem Schreiben Privilegien oder höher.

1. Gehe zum ExtraHop Codebeispiele GitHub-Repository und laden Sie das herunter `add_cloud_props_lambda/add_cloud_props_lambda.py` Datei auf Ihrem lokalen Computer.
2. Öffnen Sie in einem Texteditor den `add_cloud_props_lambda.py` archivieren und ersetzen Sie die folgenden Konfigurationsvariablen durch Informationen aus Ihrer Umgebung:
   - **HOSTNAME:** Die private IP-Adresse oder der Hostname der Sensor- oder Konsolen-EC2-Instance.
   - **EIN HAHN:** Der ExtraHop API-Schlüssel.
3. Füge das `add_cloud_props_lambda.py` Datei in eine Zip-Datei mit dem `requests` Python-Modul.

   Das Skript importiert die `requests` Python-Modul, das standardmäßig nicht für Lambda-Funktionen verfügbar ist. Informationen zum Erstellen einer Zip-Datei zum Importieren von Bibliotheken von Drittanbietern in Lambda finden Sie in der AWS-Dokumentation.
4. Erstellen Sie in AWS eine Lambda-Funktion.

   Weitere Informationen zum Erstellen von Lambda-Funktionen finden Sie in der AWS-Dokumentation.
5. Klicken Sie auf der Lambda-Funktionsseite auf **Aktionen** und wähle **Laden Sie eine.zip hoch** Datei.
6. Wählen Sie die von Ihnen erstellte Zip-Datei aus.

Hinzufügen von Azure-Eigenschaften zu ExtraHop mit Python

Das ExtraHop GitHub-Repository enthält ein Python-Beispielskript, das Azure-Geräteeigenschaften in das ExtraHop-System importiert. Das Skript weist jedem Gerät, das vom ExtraHop-System mit einer MAC-Adresse erkannt wird, die zu einer Azure-VM-Netzwerkschnittstelle gehört, Cloud-Geräteeigenschaften zu. Das Skript ist so konzipiert, dass es in einem festgelegten Zeitintervall ausgeführt wird. Jedes Mal, wenn das Skript ausgeführt wird, scannt es jede VM und aktualisiert die entsprechenden Geräte in ExtraHop.

Das Skript benötigt die folgenden Module aus dem Azure Python SDK:

- azure.mgmt.compute
- azure.mgmt.network
- azure.common.credentials

Für das Skript müssen Sie außerdem Azure-Anmeldeinformationen in den folgenden Umgebungsvariablen auf dem Computer konfiguriert haben, auf dem das Skript ausgeführt wird:

- AZURE_ABONNEMENT_ID
- AZURE_CLIENT-ID
- AZURE_CLIENT_SECRET
- AZURE_TENANT_ID

Informationen zum Generieren dieser Anmeldedaten finden Sie in der Azure-Dokumentation.

Wichtig: Das Beispiel-Python-Skript authentifiziert sich beim Sensor oder der Konsole über einen API-Schlüssel, der nicht mit der RevealX 360-REST-API kompatibel ist. Um dieses Skript mit RevealX 360 auszuführen, müssen Sie das Skript so ändern, dass es sich mit API-Token authentifiziert. Sehen Sie die py_rx360_auth.py Skript im ExtraHop GitHub-Repository für ein Beispiel für die Authentifizierung mit API-Token.

1. Gehe zum ExtraHop Codebeispiele GitHub-Repository und laden Sie das herunter `add_cloud_props_azure/add_cloud_props_azure.py` Datei auf Ihrem lokalen Computer.
2. Öffnen Sie in einem Texteditor den `add_cloud_props_azure.py` archivieren und ersetzen Sie die folgenden Konfigurationsvariablen durch Informationen aus Ihrer Umgebung:
   - **HOSTNAME:** Die IP-Adresse oder der Hostname des Sensor oder der Konsole.
   - **EIN HÜNENSCHLÜSSEL:** Der ExtraHop API-Schlüssel.
3. Führen Sie den folgenden Befehl aus:

```
python3 add_cloud_props_azure.py
```

Hinweis: Wenn das Skript eine Fehlermeldung zurückgibt, dass die TLS-Zertifikatsüberprüfung fehlgeschlagen ist, stellen Sie sicher, dass Ihrem Sensor oder Ihrer Konsole wurde ein vertrauenswürdiges Zertifikat hinzugefügt. Alternativ können Sie das hinzufügen `verify=False` Option zur Umgehung der Zertifikatsüberprüfung. Diese Methode ist jedoch nicht sicher und wird nicht empfohlen. Der folgende Code sendet eine HTTP GET-Anfrage ohne Zertifikatsüberprüfung:

```
requests.get(url, headers=headers, verify=False)
```