Fügen Sie Geräte-Cloud-Instanzeigenschaften über die REST-API hinzu

Veröffentlicht: 2024-03-27

Mithilfe der Cloud-Eigenschaften von Geräten können Sie Informationen über Ihre Cloud-Umgebung im ExtraHop-System anzeigen. 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 sowohl über den ExtraHop API Explorer als auch über Python-Python-Skripte für Amazon AWS und Microsoft Azure. Wenn Sie Cloud-Eigenschaften mit einem REST-API-Skript aktualisieren, können Sie kontinuierlich Informationen von Ihrem Cloud-Anbieter abrufen, um sicherzustellen, dass Ihre Cloud-Eigenschaftsinformationen immer auf dem neuesten Stand sind.

Bevor Sie beginnen

- Sie müssen sich anmelden bei Sensor oder Konsole mit einem Konto, das über volle Schreibrechte verfügt, um einen API-Schlüssel zu generieren.
- Sie benötigen einen gültigen API-Schlüssel, um Änderungen über die REST-API vornehmen und die folgenden Verfahren ausführen zu können. (siehe Generieren Sie einen API-Schlüssel.)
- Machen Sie sich vertraut mit dem ExtraHop REST API-Leitfaden um zu lernen, wie man im ExtraHop API Explorer navigiert.

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

1. Navigieren Sie in einem Browser zum ExtraHop 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. klicken **API-Schlüssel eingeben** und fügen Sie dann Ihren API-Schlüssel ein oder geben Sie ihn in das **API-Schlüssel** Feld.
3. klicken **Autorisieren** und klicken Sie dann auf **Schliessen**.
4. Finden Sie die ID des Gerät, indem Sie nach der MAC-Adresse des Gerät suchen.
   a) klicken **Gerät** und klicken Sie dann **POST /Geräte/Suche**.
   b) klicken **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 **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 **PATCH /geräte/ {id}**.
   b) klicken **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 **Anfrage senden**.

Rufen Sie das Lambda-Python-Beispielskript für AWS ab und installieren Sie es

Das ExtraHop GitHub-Repository enthält ein Python-Beispielskript, das AWS EC2-Instanzeigenschaften in das ExtraHop-System importiert. Das Skript ordnet Netzwerkschnittstellen von EC2-Instances Geräten zu, die auf dem ExtraHop-System anhand der MAC-Adresse erkannt wurden.

Das Skript ist so konzipiert, dass es als Lambda-Funktion in AWS ausgeführt werden kann. 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 in Ihrer VPC zugreifen können. Weitere Informationen finden Sie im AWS-Tutorial hier.
- Die Lambda-Funktion muss Listen- und Lesezugriff auf die Aktion DescribeInstances für den EC2-Dienst haben. Weitere Informationen finden Sie im AWS-Tutorial hier.

Wichtig: Das Python-Beispielskript authentifiziert sich beim Sensor oder der Konsole über einen API-Schlüssel, der nicht mit der Reveal (x) 360-REST-API kompatibel ist. Um dieses Skript mit Reveal (x) 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.

Hinweis: Wenn das Skript eine Fehlermeldung zurückgibt, dass die SSL-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)
```

1. Gehe zum ExtraHop Codebeispiele GitHub-Repository und laden Sie die `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 EC2-Instance des Sensor oder der Konsole.

   • **API-SCHLÜSSEL:** Der ExtraHop API-Schlüssel.
3. Füge die `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-Datei hoch** datei.
6. Wählen Sie die Zip-Datei aus, die Sie erstellt haben.

Rufen Sie das Python-Beispielskript für Azure ab und installieren Sie es

Das ExtraHop GitHub-Repository enthält ein Python-Skript, das Azure-Geräteeigenschaften in das ExtraHop-System importiert. Das Skript weist jedem vom ExtraHop-System erkannten Gerät mit einer MAC-Adresse, 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-Authentifizierungsanmeldeinformationen 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 Python-Beispielskript authentifiziert sich beim Sensor oder der Konsole über einen API-Schlüssel, der nicht mit der Reveal (x) 360-REST-API kompatibel ist. Um dieses Skript mit Reveal (x) 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 GitHub-Repository mit ExtraHop-Codebeispielen und laden Sie die `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.

   • **API-SCHLÜSSEL:** 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 SSL-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)
```