Beobachtungen über die REST-API hinzufügen

Veröffentlicht: 2024-11-03

Beobachtungen ermöglichen es Ihnen, zwei oder mehr IP-Adressen zuzuordnen. Sie können beispielsweise eine Beobachtung hinzufügen, die die Aktivität eines VPN-Benutzers verfolgt, indem Sie VPN-Protokolle lesen und dann die IP-Adresse des VPN-Clients in Ihrem Netzwerk mit der externen IP-Adresse verknüpfen, die dem Benutzer im Internet zugewiesen wurde. Dieses Handbuch enthält Anweisungen zum Hinzufügen einer Beobachtung über den ExtraHop REST API Explorer und über ein Python-Skript.

Bevor Sie beginnen

- Machen Sie sich mit dem vertraut ExtraHop REST-API-Leitfaden um zu erfahren, wie Sie im ExtraHop REST API Explorer navigieren.
- Für Sensoren und ECA-VMs benötigen Sie einen gültigen API-Schlüssel, um Änderungen über die REST-API vorzunehmen und die folgenden Verfahren durchzuführen. (siehe Generieren Sie einen API-Schlüssel.)
- Für RevealX 360 benötigen Sie gültige REST-API-Anmeldeinformationen, um Änderungen über die REST-API vorzunehmen und die folgenden Verfahren durchzuführen. (siehe REST-API-Anmeldeinformationen erstellen.)

Fügen Sie Beobachtungen über den REST API Explorer hinzu

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. Klicken Sie **Beobachtungen** und klicken Sie dann **POST /observations/associatedipaddrs**.
5. Klicken Sie **Probiere es aus**.

   Das JSON-Schema wird automatisch zum Textfeld für den Body-Parameter hinzugefügt.
6. Geben Sie im Textfeld Haupttext die Beobachtungen an, die Sie hinzufügen möchten.

   Die folgenden Felder verknüpfen beispielsweise 10.8.0.0 mit 108.162.0.0:

```
{
    "observations": [
        {
            "associated_ipaddr": "108.162.0.0",
            "ipaddr": "10.8.0.0",
            "timestamp": 1257935231
        }
    ],
    "source": "OpenVPN"
}
```

7. Klicken Sie **Anfrage senden**.

Rufen Sie das Python-Beispielskript ab und führen Sie es aus

Das ExtraHop GitHub-Repository enthält ein Python-Beispielskript, das Verknüpfungen auf dem ExtraHop-System auf der Grundlage einer CSV-Protokolldatei von OpenVPN erstellt. Sie können das Skript so konfigurieren, dass es andere CSV-Dateien liest, indem Sie die `IPADDR`, `ASSOCIATED_IPADDR`, und `TIMESTAMP` Variablen, die die Namen der CSV-Spalten angeben, die das Skript liest.

1. Gehe zum GitHub-Repository mit ExtraHop-Codebeispielen und laden Sie die `add_observations/add_observations.py` Datei auf Ihrem lokalen Computer.
2. Öffnen Sie in einem Texteditor den `add_observations.py` archivieren und ersetzen Sie die folgenden Konfigurationsvariablen durch Informationen aus Ihrer Umgebung:

   • **GASTGEBER:** Die IP-Adresse oder der Hostname des Sensor.

   • **API_KEY:** Der API-Schlüssel.

   • **CSV_DATEI:** Der Name der CSV-Protokolldatei.

   • **QUELLE:** Die Quelle der Beobachtungen.

   • **IPADDR:** Der Name der Spalte in der CSV-Datei, die die IP-Adressen der VPN-Clients in Ihrem internen Netzwerk angibt.

   • **ASSOZIIERTE_IPADDR:** Der Name der Spalte in der CSV-Datei, die die externen IP-Adressen angibt, die den Benutzern im öffentlichen Internet zugewiesen wurden.

   • **ZEITSTEMPEL:** Der Name der Spalte in der CSV-Datei, die den Zeitpunkt angibt, zu dem die Beobachtung von der Quelle erstellt wurde. Standardmäßig muss der Zeitstempel das folgende Format haben: `Month/Day/Year Hour:Minute:Second`. Sie können das Format jedoch ändern, indem Sie die `pattern` variabel in der `translateTime()` Funktion.

   Hinweis: Wenn die Protokolldatei Zeitstempelwerte über mehrere Spalten verteilt, können Sie die `timestamp` Feld in der `readCSV()` Funktion, um die Werte zu verketten. Nehmen wir beispielsweise an, dass die ersten vier Spalten der CSV-Datei wie in der folgenden Tabelle dargestellt angeordnet sind:

Monat	Tag	Jahr	Zeit
01	01	01	10:10:10

   Der folgende Code liest die ersten vier Spalten in die Standardspalten `translateTime()` funktion:

```
'timestamp': translateTime(row[0] + '/' + row[1] + '/' + row[2] +
 ' ' + row[3])
```

3. Führen Sie den folgenden Befehl aus:

```
python3 add_observations.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)
```