Extrahieren Sie Dateien aus Paketen über die REST-API

Veröffentlicht: 2025-03-28

Dieses Handbuch enthält Anweisungen zum Extrahieren von Dateien (auch bekannt als File Carving) über den ExtraHop REST API Explorer und über ein Python-Skript.

Bevor Sie beginnen

- Für Sensoren und die ExtraHop-Konsole 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.)

Extrahieren Sie Dateien über den REST API Explorer

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 die ExtraHop-Konsole klicken Sie auf **API-Schlüssel eingeben** und fügen Sie dann Ihren API-Schlüssel ein oder geben Sie ihn in das **API-Schlüssel** Feld.
   - Für RevealX 360 klicken Sie 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 **Autorisieren** und klicken Sie dann **Schliessen**.
4. klicken **Paketsuche** und klicken Sie dann **POST /Pakete/Suche**.
5. klicken **Probiere es aus**.

   Das JSON-Schema wird automatisch dem hinzugefügt **Körper** Textfeld.
6. In der **Körper** Textfeld, geben Sie Suchparameter für die Pakete an, aus denen Sie Dateien extrahieren möchten.

   Beispielsweise rufen die folgenden Parameter Dateien aus Paketen ab, die in den letzten 30 Minuten an oder von der IP-Adresse 10.10.10.10 gesendet wurden:

   ```
   {
       "from": "-30m",
       "output": "extract",
       "ip1": "10.10.10.10"
   }
   ```
7. klicken **Anfrage senden**.

   Wenn die Anfrage abgeschlossen ist, Antwort des Servers Abschnitt erscheint. Wenn die Anfrage erfolgreich war, wird ein 200-Statuscode angezeigt.
8. Klicken Sie neben dem 200-Statuscode auf **Datei herunterladen**.

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

Das ExtraHop GitHub-Repository enthält ein Python-Beispielskript, das Dateien aus Paketen über die REST-API extrahiert.

1. Gehe zum ExtraHop Codebeispiele GitHub-Repository und laden Sie das herunter `extract_files/extract_files.py` Datei auf Ihrem lokalen Computer.
2. Öffnen Sie in einem Texteditor den `extract_files.py` archivieren und ersetzen Sie die Konfigurationsvariablen durch Informationen aus Ihrer Umgebung.
   a) Geben Sie für Sensoren und die ExtraHop-Konsole die folgenden Konfigurationsvariablen an:
      - **GASTGEBER:** Die IP-Adresse oder der Hostname des Sensor oder der Konsole.
      - **API-SCHLÜSSEL:** Der API-Schlüssel.

   b) Geben Sie für Reveal (x) 360 die folgenden Konfigurationsvariablen an:
      - **GASTGEBER:** Der Hostname der Reveal (x) 360-API. Dieser Hostname wird auf der Reveal (x) 360 API Access-Seite unter API-Endpunkt angezeigt. Der Hostname enthält das /oauth2/token nicht.
      - **ID:** Die ID der Reveal (x) 360-REST-API-Anmeldeinformationen.
      - **GEHEIM:** Das Geheimnis der Reveal (x) 360 REST-API-Anmeldeinformationen.

   c) Geben Sie für alle Systeme die **SUCHE** Konfigurationsvariable für die Pakete, aus denen Sie Dateien extrahieren möchten.
3. Führen Sie den folgenden Befehl aus:

```
python3 extract_files.py
```

Wenn das System erfolgreich ist, werden die Dateien in einem gespeichert `.zip` Datei benannt `extracted_files.zip`.

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)
```