Ajouter des observations via l'API REST

Publié: 2024-03-20

Les observations vous permettent d'associer deux adresses IP ou plus. Par exemple, vous pouvez ajouter une observation qui suit l'activité d'un utilisateur VPN en lisant les journaux VPN, puis en associant l'adresse IP du client VPN de votre réseau à l'adresse IP externe attribuée à l'utilisateur sur Internet. Ce guide fournit des instructions pour ajouter une observation via l'explorateur d'API REST ExtraHop et via un script Python.

Avant de commencer

- Vous devez vous connecter au sonde avec un compte disposant de tous les privilèges d'écriture nécessaires pour générer une clé d'API.
- Vous devez disposer d'une clé d'API valide pour apporter des modifications via l'API REST et suivre les procédures ci-dessous. (Voir Génération d'une clé d'API.)
- Familiarisez-vous avec le Guide de l'API REST ExtraHop pour apprendre à naviguer dans l'explorateur d'API REST d'ExtraHop.

Ajoutez des observations via l'explorateur d'API REST

1. Dans un navigateur, accédez à l'explorateur d'API REST.
   L'URL est le nom d'hôte ou l'adresse IP de votre sonde, suivi par `/api/v1/explore/`. Par exemple, si votre nom d'hôte est seattle-eda, l'URL est `https://seattle-eda/api/v1/explore/`.
2. Cliquez **Entrez la clé API** puis collez ou saisissez votre clé API dans le **Clé API** champ.
3. Cliquez **Autoriser** puis cliquez sur **Fermer**.
4. Cliquez **Observations** puis cliquez sur **POST /observations/pagaies associées**.
5. Cliquez **Essayez-le**.
   Le schéma JSON est automatiquement ajouté à la zone de texte du paramètre du corps.
6. Dans la zone de texte du corps, indiquez les observations que vous souhaitez ajouter.
   Par exemple, les champs suivants associent 10.8.0.0 à 108.162.0.0 :

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

7. Cliquez **Envoyer la demande**.

Récupérez et exécutez l'exemple de script Python

Le référentiel GitHub d'ExtraHop contient un exemple de script Python qui crée des associations sur le système ExtraHop sur la base d'un fichier journal CSV d'OpenVPN. Vous pouvez configurer le script pour lire d'autres fichiers CSV en modifiant `IPADDR`, `ASSOCIATED_IPADDR`, et `TIMESTAMP` variables, qui spécifient les noms des colonnes CSV lues par le script.

1. Accédez au Référentiel GitHub d'exemples de code ExtraHop et téléchargez le `add_observations/add_observations.py` fichier sur votre machine locale.

2. Dans un éditeur de texte, ouvrez `add_observations.py` archivez et remplacez les variables de configuration suivantes par des informations provenant de votre environnement :

   • **HÔTE:** L'adresse IP ou le nom d'hôte de la sonde.

   • **CLÉ_API:** La clé API.

   • **FICHIER_CSV:** Nom du fichier journal CSV.

   • **SOURCE:** La source des observations.

   • **IPADDR:** Nom de la colonne du fichier CSV qui indique les adresses IP des clients VPN de votre réseau interne.

   • **ASSOCIATED_IPADDR:** Nom de la colonne du fichier CSV qui indique les adresses IP externes attribuées aux utilisateurs sur l'Internet public.

   • **HORODATAGE:** Nom de la colonne du fichier CSV qui indique l'heure à laquelle l'observation a été créée par la source. Par défaut, l' horodateur doit être au format suivant : `Month/Day/Year Hour:Minute:Second`. Toutefois, vous pouvez modifier le format en modifiant le `pattern` variable dans le `translateTime()` fonction.

   Conseil: Si le fichier journal distribue les valeurs d'horodateur sur plusieurs colonnes , vous pouvez modifier le `timestamp` champ dans le `readCSV()` fonction pour concaténer les valeurs. Supposons, par exemple, que les quatre premières colonnes du fichier CSV soient organisées comme indiqué dans le tableau suivant :

Mois	Journée	Année	Heure
01	01	01	10:10:10

   Le code suivant lit ces quatre premières colonnes dans la valeur par défaut `translateTime()` fonction :

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

3. Exécutez la commande suivante :

```
python3 add_observations.py
```

   Note: Si le script renvoie un message d'erreur indiquant que la vérification du certificat SSL a échoué, assurez-vous que un certificat fiable a été ajouté à votre sonde ou à votre console. Vous pouvez également ajouter le `verify=False` option permettant de contourner la vérification des certificats. Cependant, cette méthode n'est pas sûre et n'est pas recommandée. Le code suivant envoie une requête HTTP GET sans vérification du certificat :

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