

Créez des appareils personnalisés via l'API REST

Publié: 2023-10-24

Vous pouvez créer des appareils personnalisés via l'API REST qui suit le trafic réseau sur plusieurs adresses IP et ports. Par exemple, vous souhaitez peut-être ajouter un équipement personnalisé pour chaque succursale. Si vous créez les appareils par le biais d'un script, vous pouvez lire la liste des appareils à partir d'un fichier CSV. Dans cette rubrique, nous allons présenter des méthodes pour l'API REST et pour l'explorateur d'API REST ExtraHop.

Avant de commencer

- Vous devez vous connecter au sonde avec un compte doté de privilèges d'administration du système et d'accès 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.

Création d'un équipement personnalisé via l'explorateur d'API REST

Vous pouvez créer un équipement personnalisé et l'associer à une liste d' adresses IP ou de blocs CIDR par le biais du **POST/appareils personnalisés** opération.

! Important: L'explorateur d'API REST n'est pas disponible sur Reveal (x) 360.

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 de `/api/v1/explore/`. Par exemple, si votre nom d'hôte est `seattle-eda`, l'URL est `https://seattle-eda/api/v1/explore/`.
2. Cliquez **Appareil personnalisé**, puis cliquez sur **POST/appareils personnalisés**.
3. Dans le champ body, spécifiez les propriétés de l'équipement personnalisé que vous souhaitez créer.
Par exemple, le corps suivant fait correspondre l'équipement personnalisé aux blocs CIDR `192.168.0.0/26`, `192.168.0.64/27`, `192.168.0.96/30` et `192.168.0.100/32` :

```
{
  "description": "The location of our office in Washington",
  "name": "Seattle",
  "criteria": [
    {
      "ipaddr": "192.168.0.0/26"
    },
    {
      "ipaddr": "192.168.0.64/27"
    },
    {
      "ipaddr": "192.168.0.96/30"
    },
    {
      "ipaddr": "192.168.0.100/32"
    }
  ]
}
```

Récupérez et exécutez l'exemple de script Python

Le référentiel GitHub d'ExtraHop contient un exemple de script Python qui crée des appareils personnalisés en lisant les critères d'un fichier CSV.

1. Accédez au [Référentiel GitHub d'exemples de code ExtraHop](#) et téléchargez le `create_custom_devices/create_custom_devices.py` fichier sur votre machine locale.
2. Créez un fichier CSV avec des lignes contenant les colonnes suivantes dans l'ordre indiqué :

Nom	IDENTIFIANT	Descriptif	adresse IP ou bloc CIDR
-----	-------------	------------	-------------------------



Conseil `create_custom_devices` le répertoire contient un exemple de fichier CSV nommé `device_list.csv`.

Le script n'accepte pas de ligne d'en-tête dans le fichier CSV. Le nombre de colonnes du tableau n'est pas limité ; chaque colonne située après les quatre premières indique une adresse IP supplémentaire pour l'équipement. Les quatre premières colonnes sont obligatoires pour chaque ligne.

3. Dans un éditeur de texte, ouvrez le `create_custom_devices.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.
 - **APIKEY**: La clé d'API.
 - **FICHIER_CSV**: Le chemin du fichier CSV par rapport à l'emplacement du fichier de script.
4. Exécutez la commande suivante :

```
python3 create_custom_devices.py
```



Note: Si le script renvoie un message d'erreur indiquant que la vérification du certificat SSL a échoué, assurez-vous que [un certificat de confiance a été ajouté à votre sonde ou à votre console](#). Vous pouvez également ajouter le `verify=False` possibilité 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 de certificat :

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