Migrer les règles de réglage

Publié: 2024-03-27

Vous pouvez migrer les règles de réglage à partir d'une sonde ou console à un autre via l'API REST. Cela peut être utile si vous avez créé un grand nombre de règles de réglage et que vous ne souhaitez pas les recréer manuellement. Dans cette rubrique, nous présentons les méthodes permettant de migrer une règle manuellement via l'explorateur d'API REST et de migrer les règles à l'aide de scripts Python. Un exemple de script fait migrer les règles entre deux machines virtuelles ECA et un exemple de script fait migrer les règles d'une machine virtuelle ECA vers Reveal (x) 360.

Avant de commencer

- Les deux capteurs ou consoles doivent exécuter la version 8.4 ou ultérieure du microprogramme.
- Si vous migrez des règles de réglage qui font référence à des groupes d'équipements, envisagez de migrer ces groupes d' équipements par le biais d'un bundle. Tu peux créer un bundle avec les groupes d'équipements du système source et installer le bundle sur le système cible.

Migrer une règle de réglage via l'explorateur d'API REST

1. Récupérez les métadonnées des règles de réglage depuis le système source.
   a) Dans un navigateur, accédez à l'explorateur d'API REST.

      L'URL est le nom d'hôte ou l'adresse IP de votre sonde ou console, suivi par `/api/v1/explore/`. Par exemple, si votre nom d'hôte est seattle-eda, l'URL est `https://seattle-eda/api/v1/explore/`.
   b) Cliquez **Entrez la clé API** puis collez ou saisissez votre clé API dans le **Clé API** champ.
   c) Cliquez **Autoriser** puis cliquez sur **Fermer**.
   d) Cliquez **Détections**.
   e) Cliquez **GET /detections/rules/masquage**.
   f) Cliquez **Essayez-le**.
   g) Cliquez **Envoyer la demande**.
   h) Dans le champ Corps de la réponse, copiez l'objet JSON qui représente la règle de réglage que vous souhaitez copier.
2. Recréez la règle de réglage sur le système cible.
   a) Dans un navigateur, accédez à l'explorateur d'API REST.
   b) Cliquez **Entrez la clé API** puis collez ou saisissez votre clé API dans le **Clé API** champ.
   c) Cliquez **Autoriser** puis cliquez sur **Fermer**.
   d) Cliquez **Détections**.
   e) Cliquez **POST /détections/règles/masquage**.
   f) Cliquez **Essayez-le**.
   g) Dans la zone de texte du corps, collez l'objet JSON que vous avez copié depuis la source sonde ou console.

      L'entrée doit ressembler au texte suivant :

```
{
    "id": 1,
    "enabled": false,
    "detection_type": "cifs_round_trip_time",
    "offender": {
      "object_type": "device",
      "object_id": 123
    },
    "victim": {
      "object_type": "device",
```

```
        "object_id": 321
    },
    "author": "example_user",
    "create_time": 1615588932838,
    "expiration": 1615675096000,
    "detections_hidden": 0
}
```

Note: Si le `description` ou `properties` le champ est défini sur null `null`, vous devez supprimer ces champs du JSON avant d'envoyer la demande.

h) Cliquez **Envoyer la demande**.

3. Optionnel : Désactivez la règle de réglage sur le système cible.

   Si la règle de réglage a été désactivée sur le système source, indiqué par le champ activé défini sur `false`, définissez le champ activé sur False sur le système cible.

   a) Dans un navigateur, accédez à l'explorateur d'API REST.
   b) Cliquez **Entrez la clé API** puis collez ou saisissez votre clé API dans le **Clé API** champ.
   c) Cliquez **Autoriser** puis cliquez sur **Fermer**.
   d) Cliquez **Détections**.
   e) Cliquez **PATCH /détections/règles/masquage**.
   f) Cliquez **Essayez-le**.
   g) Dans le corps du texte, collez le code JSON suivant :

```
{
    "enabled": false
}
```

   h) Cliquez **Envoyer la demande**.

Récupérez et exécutez l'exemple de script Python pour Reveal (x) 360

Le référentiel GitHub ExtraHop contient un exemple de script Python qui migre toutes les règles d'exceptions d'une machine virtuelle ECA vers Reveal (x) 360.

Note: Le script migre uniquement les règles qui sont activées.

1. Accédez au Référentiel GitHub d'exemples de code ExtraHop et téléchargez le `migrate_detection_hiding/migrate_detection_hiding.py` fichier sur votre machine locale.
2. Dans un éditeur de texte, ouvrez `migrate_detection_hiding.py` archivez et remplacez les variables de configuration suivantes par des informations provenant de votre environnement :

   • **HÔTE SOURCE:** Le nom d'hôte de la machine virtuelle ECA à partir de laquelle vous migrez les règles d'exceptions

   • **CLÉ_SOURCE DE L'API:** La clé API de la machine virtuelle ECA à partir de laquelle vous migrez les règles d'exceptions

   • **HÔTE_CIBLE:** Le nom d'hôte de l'API Reveal (x) 360 vers laquelle vous migrez les règles d' exceptions. Ce nom d'hôte est affiché sur la page d'accès à l'API Reveal (x) 360 sous API Endpoint. Le nom d'hôte n'inclut pas `/oauth2/token`.

   • **IDENTIFIANT_CIBLE:** L'ID des informations dcessaires d'identification de l'API REST pour Reveal (x) 360

   • **CIBLE_SECRET:** Le secret des informations dcessaires d'identification de l'API REST pour Reveal (x) 360

3. Exécutez la commande suivante :

```
python3 migrate_detection_hiding.py
```

Si les règles d'exceptions spécifient les appareils participants ou les groupes d'appareils par un ID, le script essaie de trouver les identifiants des participants équivalents sur Reveal (x) 360 en recherchant les adresses IP des appareils et les noms des groupes d'appareils.

Si le script ne trouve pas les identifiants des participants équivalents sur Reveal (x) 360, il vous invite à migrer les autres règles pour lesquelles des participants équivalents ont été trouvés. Pour continuer, tapez y et appuyez sur ENTER.

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)
```

Récupérez et exécutez l'exemple de script Python pour Reveal (x) Enterprise

Le référentiel GitHub ExtraHop contient un exemple de script Python qui migre toutes les règles de réglage d'une machine virtuelle ECA vers une autre machine virtuelle ECA.

Note: Le script migre uniquement les règles activées.

1. Accédez au Référentiel GitHub d'exemples de code ExtraHop et téléchargez le `migrate_detection_hiding/migrate_detection_hiding_enterprise.py` fichier sur votre machine locale.
2. Dans un éditeur de texte, ouvrez le `migrate_detection_hiding_enterprise.py` archivez et remplacez les variables de configuration suivantes par des informations provenant de votre environnement :

   • **SOURCE_HÔTE:** Le nom d'hôte de la machine virtuelle ECA à partir de laquelle vous migrez les règles de réglage

   • **SOURCE_API_KEY:** La clé d'API de la machine virtuelle ECA à partir de laquelle vous migrez les règles de réglage

   • **HÔTE_CIBLE:** Le nom d'hôte de la machine virtuelle ECA vers laquelle vous migrez les règles de réglage

   • **CLÉ_API CIBLE:** La clé d'API de la machine virtuelle ECA vers laquelle vous migrez les règles de réglage
3. Exécutez la commande suivante :

```
python3 migrate_detection_hiding_enterprise.py
```

Si les règles de réglage spécifient les appareils participants ou les groupes d'équipements par un identifiant, le script essaie de trouver les identifiants des participants équivalents sur la machine virtuelle ECA cible en recherchant les adresses IP des équipements et les noms de groupes de périphériques.

Si le script ne trouve pas les identifiants des participants équivalents sur la machine virtuelle ECA cible, le script vous invite à migrer les autres règles pour lesquelles des participants équivalents ont été trouvés. Pour continuer, tapez y et appuyez sur ENTER.

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)
```