Tuning-Regeln migrieren

Veröffentlicht: 2024-07-02

Sie können Tuning-Regeln von einem migrieren Sensor oder Konsole zu einem anderen über die REST-API. Dies kann nützlich sein, wenn Sie eine große Anzahl von Optimierungsregeln erstellt haben und diese nicht manuell neu erstellen möchten. In diesem Thema zeigen wir Methoden zum manuellen Migrieren einer Regel über den REST API Explorer und zum Migrieren von Regeln mit Python-Skripten. Ein Beispielskript migriert Regeln zwischen zwei ECA-VMs und ein Beispielskript migriert Regeln von einer ECA-VM zu RevealX 360.

Bevor Sie beginnen

- Auf beiden Sensoren oder Konsolen muss die Firmware-Version 8.4 oder höher ausgeführt werden.
- Wenn Sie Tuning-Regeln migrieren, die auf Gerätegruppen verweisen, sollten Sie erwägen, diese Gerätegruppen mit einem Paket zu migrieren. Du kannst ein Paket erstellen mit den Gerätegruppen auf dem Quellsystem und installiere das Paket auf dem Zielsystem.

Migrieren Sie eine Optimierungsregel über den REST API Explorer

1. Ruft die Tuning-Regel-Metadaten aus dem Quellsystem ab.
   a) 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/`.
   b) klicken **API-Schlüssel eingeben** und fügen Sie dann Ihren API-Schlüssel ein oder geben Sie ihn in das **API-Schlüssel** Feld.
   c) klicken **Autorisieren** und klicken Sie dann **Schliessen**.
   d) klicken **Erkennungen**.
   e) klicken **GET /Erkennungen/Regeln/Verbergen**.
   f) klicken **Probiere es aus**.
   g) klicken **Anfrage senden**.
   h) Kopieren Sie im Feld Antworttext das JSON-Objekt, das die Optimierungsregel darstellt, die Sie kopieren möchten.
2. Erstellen Sie die Tuning-Regel auf dem Zielsystem neu.
   a) Navigieren Sie in einem Browser zum REST API Explorer.
   b) klicken **API-Schlüssel eingeben** und fügen Sie dann Ihren API-Schlüssel ein oder geben Sie ihn in das **API-Schlüssel** Feld.
   c) klicken **Autorisieren** und klicken Sie dann **Schliessen**.
   d) klicken **Erkennungen**.
   e) klicken **POST /Erkennungen/Regeln/Verbergen**.
   f) klicken **Probiere es aus**.
   g) Fügen Sie das JSON-Objekt, das Sie aus der Quelle kopiert haben, in das Textfeld ein Sensor oder Konsole.

      Der Eintrag sollte dem folgenden Text ähneln:

```
{
    "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
}
```

Hinweis: Wenn der `description` oder `properties` Feld ist auf Null gesetzt `null`, Sie müssen diese Felder aus dem JSON entfernen, bevor Sie die Anfrage senden.

h) klicken **Anfrage senden**.

3. Optional: Deaktivieren Sie die Tuning-Regel auf dem Zielsystem.

   Wenn die Optimierungsregel auf dem Quellsystem deaktiviert war, wird dies durch das aktivierte Feld angezeigt, das auf gesetzt ist `false`, setzen Sie das aktivierte Feld auf dem Zielsystem auf falsch.

   a) Navigieren Sie in einem Browser zum REST API Explorer.
   b) klicken **API-Schlüssel eingeben** und fügen Sie dann Ihren API-Schlüssel ein oder geben Sie ihn in das **API-Schlüssel** Feld.
   c) klicken **Autorisieren** und klicken Sie dann **Schliessen**.
   d) klicken **Erkennungen**.
   e) klicken **PATCH /Erkennungen/Regeln/Verbergen**.
   f) klicken **Probiere es aus**.
   g) Fügen Sie in das Textfeld den folgenden JSON-Code ein:

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

   h) klicken **Anfrage senden**.

Rufen Sie das Python-Beispielskript für RevealX 360 ab und führen Sie es aus

Das ExtraHop GitHub-Repository enthält ein Python-Beispielskript, das alle Tuning-Regeln auf einer ECA-VM zu RevealX 360 migriert.

Hinweis: Das Skript migriert nur Regeln, die aktiviert sind.

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

   • **QUELLE_HOST:** Der Hostname der ECA-VM, von der Sie Tuning-Regeln migrieren

   • **QUELLE_API_KEY:** Der API-SCHLÜSSEL auf der ECA-VM, von der Sie Tuning-Regeln migrieren

   • **ZIELHOST:** Der Hostname der RevealX 360-API, zu der Sie Tuning-Regeln migrieren. Dieser Hostname wird auf der RevealX 360 API Access-Seite unter API-Endpunkt angezeigt. Der Hostname enthält nicht die `/oauth2/token`.

   • **ZIEL-ID:** Die ID der REST-API-Anmeldeinformationen für RevealX 360

   • **TARGET_SECRET:** Das Geheimnis der REST-API-Anmeldeinformationen für RevealX 360

3. Führen Sie den folgenden Befehl aus:

```
python3 migrate_detection_hiding.py
```

Wenn Tuning-Regeln Teilnehmergeräte oder Gerätegruppen anhand einer ID spezifizieren, versucht das Skript, die IDs gleichwertiger Teilnehmer auf RevealX 360 zu finden, indem es nach Geräte-IP-Adressen und Gerätegruppennamen sucht.

Wenn das Script die IDs für gleichwertige Teilnehmer auf RevealX 360 nicht finden kann, fordert das Script Sie auf, die anderen Regeln zu migrieren, für die gleichwertige Teilnehmer gefunden wurden. Um fortzufahren, geben Sie y und drücken Sie ENTER.

Hinweis: Wenn das Skript eine Fehlermeldung zurückgibt, dass die SSL-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)
```

Rufen Sie das Python-Beispielskript für RevealX Enterprise ab und führen Sie es aus

Das ExtraHop GitHub-Repository enthält ein Python-Beispielskript, das alle Tuning-Regeln von einer ECA-VM auf eine andere ECA-VM migriert.

Hinweis: Das Skript migriert nur Regeln, die aktiviert sind.

1. Gehe zum ExtraHop Codebeispiele GitHub-Repository und laden Sie das herunter `migrate_detection_hiding/migrate_detection_hiding_enterprise.py` Datei auf Ihrem lokalen Computer.
2. Öffnen Sie in einem Texteditor den `migrate_detection_hiding_enterprise.py` archivieren und ersetzen Sie die folgenden Konfigurationsvariablen durch Informationen aus Ihrer Umgebung:
   - **QUELLE_HOST:** Der Hostname der ECA-VM, von der Sie Tuning-Regeln migrieren
   - **QUELLE_API_KEY:** Der API-SCHLÜSSEL auf der ECA-VM, von der Sie Tuning-Regeln migrieren
   - **ZIELHOST:** Der Hostname der ECA-VM, zu der Sie Tuning-Regeln migrieren
   - **ZIEL-API-SCHLÜSSEL:** Der API-KEY auf der ECA-VM, zu der Sie Tuning-Regeln migrieren
3. Führen Sie den folgenden Befehl aus:

```
python3 migrate_detection_hiding_enterprise.py
```

Wenn Tuning-Regeln Teilnehmergeräte oder Gerätegruppen anhand einer ID angeben, versucht das Skript, die IDs der entsprechenden Teilnehmer auf der ECA-Ziel-VM zu finden, indem es nach Geräte-IP-Adressen und Gerätegruppennamen sucht.

Wenn das Skript die IDs für gleichwertige Teilnehmer auf der ECA-Ziel-VM nicht finden kann, fordert das Skript Sie auf, die anderen Regeln zu migrieren, für die entsprechende Teilnehmer gefunden wurden. Um fortzufahren, geben Sie y und drücken Sie ENTER.

Hinweis: Wenn das Skript eine Fehlermeldung zurückgibt, dass die SSL-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)
```