Migrate tuning rules

Published: 2024-02-23

You can migrate tuning rules from one sensor or console to another through the REST API. This can be useful if you have created a large number of tuning rules and do not want to manually recreate them. In this topic, we show methods for migrating a rule manually through the REST API Explorer and migrating rules with Python scripts. One example script migrates rules between two ECA VMs and one example script migrates rules from an ECA VM to Reveal(x) 360.

Before you begin

- Both sensors or consoles must be running firmware version 8.4 or later.
- If you are migrating tuning rules that reference device groups, consider migrating those device groups with a bundle. You can create a bundle ⊠ with the device groups on the source system and install the bundle ⊠ on the target system.

Migrate a tuning rule through the REST API Explorer

1. Retrieve the tuning rule metadata from the source system.
   a) In a browser, navigate to the REST API Explorer.

      The URL is the hostname or IP address of your sensor or console, followed by `/api/v1/explore/`. For example, if your hostname is seattle-eda, the URL is `https://seattle-eda/api/v1/explore/`.
   b) Click **Enter API Key** and then paste or type your API key into the **API Key** field.
   c) Click **Authorize** and then click **Close**.
   d) Click **Detections**.
   e) Click **GET /detections/rules/hiding**.
   f) Click **Try it out**.
   g) Click **Send Request**.
   h) In the Response body field, copy the JSON object that represents the tuning rule you want to copy.

2. Recreate the tuning rule on the target system.
   a) In a browser, navigate to the REST API Explorer.
   b) Click **Enter API Key** and then paste or type your API key into the **API Key** field.
   c) Click **Authorize** and then click **Close**.
   d) Click **Detections**.
   e) Click **POST /detections/rules/hiding**.
   f) Click **Try it out**.
   g) In the body text box, paste the JSON object you copied from the source sensor or console.

      The entry should look similar to the following text:

```
{
    "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:** If the `description` or `properties` field is set to null `null`, you must remove those fields from the JSON before sending the request.

    h)   Click **Send Request**.

3.   Optional: Disable the tuning rule on the target system.

    If the tuning rule was disabled on the source system, indicated by the enabled field set to `false`, set the enabled field to false on the target system.

    a)   In a browser, navigate to the REST API Explorer.
    b)   Click **Enter API Key** and then paste or type your API key into the **API Key** field.
    c)   Click **Authorize** and then click **Close**.
    d)   Click **Detections**.
    e)   Click **PATCH /detections/rules/hiding**.
    f)   Click **Try it out**.
    g)   In the body text box, paste the following JSON:

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

    h)   Click **Send Request**.

Retrieve and run the example Python script for Reveal(x) 360

The ExtraHop GitHub repository contains an example Python script that migrates all tuning rules on an ECA VM to Reveal(x) 360.

> **Note:** The script only migrates rules that are enabled.

1.   Go to the ExtraHop code-examples GitHub repository ⧉ and download the `migrate_detection_hiding/migrate_detection_hiding.py` file to your local machine.

2.   In a text editor, open the `migrate_detection_hiding.py` file and replace the following configuration variables with information from your environment:

    • **SOURCE_HOST:** The hostname of the ECA VM you are migrating tuning rules from

    • **SOURCE_API_KEY:** The API KEY on the ECA VM you are migrating tuning rules from

    • **TARGET_HOST:** The hostname of the Reveal(x) 360 API you are migrating tuning rules to. This hostname is displayed in the Reveal(x) 360 API Access page under API Endpoint. The hostname does not include the `/oauth2/token`.

    • **TARGET_ID:** The ID of the REST API credentials for Reveal(x) 360

    • **TARGET_SECRET:** The secret of the REST API credentials for Reveal(x) 360

3.   Run the following command:

```
python3 migrate_detection_hiding.py
```

If tuning rules specify participant devices or device groups by an ID, the script tries to find the IDs of equivalent participants on Reveal(x) 360 by searching for device IP addresses and device group names.

If the script cannot find the IDs for equivalent participants on Reveal(x) 360, the script prompts you to migrate the other rules that equivalent participants were found for. To continue, type `y` and press ENTER.

> **Note:** If the script returns an error message that the SSL certificate verification failed, make sure that a trusted certificate has been added to your sensor or console ⬀. Alternatively, you can add the `verify=False` option to bypass certificate verification. However, this method is not secure and not recommended. The following code sends an HTTP GET request without certificate verification:
>
> ```
> requests.get(url, headers=headers, verify=False)
> ```

Retrieve and run the example Python script for Reveal(x) Enterprise

The ExtraHop GitHub repository contains an example Python script that migrates all tuning rules from an ECA VM to another ECA VM.

> **Note:** The script only migrates rules that are enabled.

1. Go to the ExtraHop code-examples GitHub repository ⬀ and download the `migrate_detection_hiding/migrate_detection_hiding_enterprise.py` file to your local machine.
2. In a text editor, open the `migrate_detection_hiding_enterprise.py` file and replace the following configuration variables with information from your environment:
    - **SOURCE_HOST:** The hostname of the ECA VM you are migrating tuning rules from
    - **SOURCE_API_KEY:** The API KEY on the ECA VM you are migrating tuning rules from
    - **TARGET_HOST:** The hostname of the ECA VM you are migrating tuning rules to
    - **TARGET_API_KEY:** The API KEY on the ECA VM you are migrating tuning rules to
3. Run the following command:

    ```
    python3 migrate_detection_hiding_enterprise.py
    ```

    If tuning rules specify participant devices or device groups by an ID, the script tries to find the IDs of equivalent participants on the target ECA VM by searching for device IP addresses and device group names.

    If the script cannot find the IDs for equivalent participants on the target ECA VM, the script prompts you to migrate the other rules that equivalent participants were found for. To continue, type `y` and press ENTER.

    > **Note:** If the script returns an error message that the SSL certificate verification failed, make sure that a trusted certificate has been added to your sensor or console ⬀. Alternatively, you can add the `verify=False` option to bypass certificate verification. However, this method is not secure and not recommended. The following code sends an HTTP GET request without certificate verification:
    >
    > ```
    > requests.get(url, headers=headers, verify=False)
    > ```