Tag a device through the REST API

Published: 2024-04-08

Tags can help you classify devices that share a common characteristic from among potentially hundreds of discovered devices on your network.

You might want to tag devices by their role on your network, such as the devices that make up your development and production servers. For example, if you are running multiple AWS instances in your environment, it is critical to size them for their workload. An undersized instance can result in poor performance; an oversized instance is needlessly costly. If you tag your AWS instances, you can easily set up device groups by instance size, and then create a dashboard to monitor usage and performance metrics.

In this guide, you will learn how create a tag, find the device you want to tag, and then add the tag to the device. An example script is provided at the end, which adds a given device tag to all IP addresses read from a CSV file.

Before you begin

- For sensors and ECA VMs, you must have a valid API key to make changes through the REST API and complete the procedures below. (See Generate an API key.)
- For Reveal(x) 360, you must have valid REST API credentials to make changes through the REST API and complete the procedures below. (See Create REST API credentials.)

Create a tag

If you already have a tag on the system, you can skip this step. The example script at the bottom of this guide will check for a tag and create a new tag only if necessary.

1. 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/`.
2. Click **Enter API Key** and then paste or type your API key into the **API Key** field.
3. Click **Authorize** and then click **Close**.
4. Click **Tag** and then click **POST/tags**.
5. Click **Try it out**.
   The JSON schema is automatically added to the body parameter text box.
6. In the `name` field, replace `string` with the new tag name.
7. Click **Send Request** to create the tag.

Retrieve devices that match your criteria

In this step you will search for the devices that you want to tag and note the device ID. You must have the device ID before you can tag devices.

1. Scroll up the page and click **Device** to display device operations.
2. Click **POST /devices/search**.
3. Click **Try it out**.
   The JSON schema is automatically added to the body parameter text box.
4. In the body text box, type search criteria that selects the devices.
   The following search criteria returns a device with an IP address of 10.10.10.200:

```
{
```

```
  "filter": {
    "field": "ipaddr",
    "operand": "10.10.10.200",
    "operator": "="
  }
}
```

For more information about device search filters, see Operand values for device search.

5. Click **Send Request**.

   In the Server response section, the Response body displays information about each device that matches your search criteria, including the device ID.

Assign the tag to a device

In this step, you will assign a tag to a device by the device ID you found in the previous step.

1. Scroll down the page and click **Tag** to display tag operations.
2. Click **POST /tags/{id}/devices/{child-id}**.
3. Click **Try it out**.
4. In the child-id field, type the ID of the device you want to tag.
5. In the id field, type the ID of the tag you want to assign.
6. Click **Send Request** to assign the tag to the device.

Tip: After you click **Send Request**, you can click the tabs to view scripts for the operation in Curl, Python 2.7, or Ruby.

Retrieve and run the example Python script

The ExtraHop GitHub repository contains an example Python script that creates a device tag and then assigns the tag to all devices with the IP addresses specified in a CSV file. The script creates a new tag only if the specified tag does not already exist.

1. Go to the ExtraHop code-examples GitHub repository and download the `tag_device/tag_device.py` file to your local machine.
2. In a text editor, open the `tag_device.py` file and replace the following configuration variables with information from your environment:
   - For sensors and ECA VMs, specify the following configuration variables:
     - **HOST:** The IP address or hostname of the sensor or ECA VM.
     - **API_KEY:** The API key.
     - **TAG:** The name of the tag
     - **DEVICE_LIST:** The file that contains the list of IP addresses
   - For Reveal(x) 360, specify the following configuration variables:
     - **HOST:** The hostname of the Reveal(x) 360 API. This hostname is displayed in the Reveal(x) 360 API Access page under API Endpoint. The hostname does not include the `/oauth2/token`.
     - **ID:** The ID of the Reveal(x) 360 REST API credentials.
     - **SECRET:** The secret of the Reveal(x) 360 REST API credentials.
     - **TAG:** The name of the tag
     - **DEVICE_LIST:** The file that contains the list of IP addresses

3. Run the following command:

```
python3 tag_device.py
```

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)
```