ExtraHop
Upload STIX files through the REST API

Published: 2022-05-25

Published: 2022-05-25

Published: 2022-05-25

Threat collections enable your ExtraHop system to identify suspicious IP addresses, hostnames, and URIs found in your network activity. While ExtraHop-curated threat collections are enabled by default, you can also upload a custom threat collection from free or commercial sources.

Before you begin

- You must have unlimited privileges to generate an API key.
- You must have a valid API key to make changes through the REST API and complete the procedures below. (See Generate an API key.)
- Familiarize yourself with threat intelligence.
- Familiarize yourself with the ExtraHop REST API Guide to learn how to navigate the ExtraHop REST API Explorer.

Threat collections must be added and updated to all connected sensors and consoles. And because these sources are often updated frequently, the REST API provides the opportunity to automate updates for threat collections to all sensors and consoles.

Custom threat collections must be formatted in Structured Threat Information Expression (STIX) as TAR files. ExtraHop systems currently support STIX versions 1.0 - 1.2.

Retrieve and run the example Python script

The ExtraHop GitHub repository contains an example Python script that uploads all STIX files in a given directory to a list of sensors and consoles. First, the script reads through a CSV file that contains the URLs and API keys for each system. For each system, the script gets a list of all threat collections that are already on the system. The script then processes each STIX file in the directory for each system.

If the name of the file matches the name of a threat collection on the system, the script overwrites the threat collection with the file contents. If there are no threat collection names that match the file name, the script uploads the file to create a new threat collection.

Important: The example python script authenticates to the sensor or console through an API key, which is not compatible with the Reveal(x) 360 REST API. To run this script with Reveal(x) 360, you must modify the script to authenticate with API tokens. See the py_rx360_auth.py script in the ExtraHop GitHub repository for an example of how to authenticate with API tokens.

1. Go to the ExtraHop code-examples GitHub repository and download the `upload_stix/upload_stix.py` file to your local machine.
2. Create a CSV file with rows that contain the following columns in the specified order:

System hostname	API key

Tip: The `upload_stix` directory contains an example CSV file named `systems.csv`.

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

- **SYSTEM_LIST:** The path of the CSV file with the HTTPS URLs and API keys of the systems
- **STIX_DIR:** The path of the directory that contains the STIX files

4. Run the following command:

```
python3 upload_stix.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)
```