Update network localities

Published: 2022-12-29

You can add multiple CIDR blocks and IP addresses to a single network locality, and you can configure a name for the locality. The ExtraHop GitHub repository contains Python scripts that help you consolidate and rename localities automatically.

> **Note:** If you created network localities in firmware previous to version 9.0, where you could only specify a single CIDR block or IP address for a network locality, you might want to consolidate and rename network localities to make searching and filtering by locality easier.

The `retrieve_network_localities.py` script retrieves all network locality information from a specified sensor or console and saves the information to a CSV file. You can modify the CSV file to specify which localities you want to consolidate and specify new names for existing localities. The `create_network_localities.py` script then reads the updated CSV file to replace the existing localities on a specified sensor or console.

> **Warning:** The `create_network_localities.py` script deletes all network localities on the target sensor or console before creating the new entries specified in the CSV file.

Consolidating network localities

In the CSV file, you can specify which localities you want to consolidate by assigning the same description to multiple localities. When the `create_network_localities.py` script consolidates the localities, it assigns the name of the first locality in the group to the new locality. For example, assume that the CSV file contains the following entries:

networks	external	description	name
192.168.1.2	False	group1	[auto]: Internal - 192.168.1.2
192.168.1.1	False	group1	[auto]: Internal - 192.168.1.1

Running the `create_network_localities.py` script creates the following network locality on the target sensor or console:

networks	external	description	name
192.168.1.2 and 192.168.1.1	False	group1	[auto]: Internal - 192.168.1.2

To consolidate network localities with the same description in the CSV file as described in this topic, you must specify the `--group description` option when you run the `create_network_localities.py` script.

Renaming network localities

In the CSV file, you can specify descriptive names for localities. The ExtraHop system automatically generates names for network localities if they are not specified by a user.

> **Note:** If you run the `retrieve_network_localities.py` script on a sensor or console running firmware version 8.9 or earlier, the script automatically generates names for each locality and adds them to the CSV file. You can change these names to

be more descriptive by changing the names in the CSV file before running the `create_network_localities.py` script.

Both the script and the ExtraHop system generate names according to the following format:

```
[auto]: EXTERNALITY - NETWORK
```

In the above text, EXTERNALITY is replaced with either "External" or "Internal", and NETWORK is replaced with the IP address or CIDR block of the network. For example, the following name is assigned to a network locality for the 192.168.1.0/24 CIDR block:

```
[auto]: Internal - 192.168.1.0/24
```

Retrieve and run the Python scripts

Note: The `create_network_localities.py` script deletes all network localities on the target sensor or console before creating the new entries specified in the CSV file.

1. Go to the ExtraHop code-examples GitHub ☑ repository and download the contents of the `update_network_localities` directory to your local machine.
2. Run the `retrieve_network_localities.py` script.
   - For sensors and ECA VMs, run the following command:

     ```
     python3 retrieve_network_localities.py HOST --apikey API_KEY
     ```

     Replace the following variables in the command with information from your ExtraHop system:

     • **HOST:** The IP address or hostname of the sensor or console.

     • **API_KEY:** The API key.
   - For Reveal(x) 360, run the following command:

     ```
     python3 retrieve_network_localities.py HOST --id ID --secret SECRET
     ```

     Replace the following variables in the command with information from your ExtraHop system:

     • **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 /oauth/token.

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

     • **SECRET:** The secret of the Reveal(x) 360 REST API credentials.

   The script saves network locality information to the `localities.csv` file in the current directory. After the file is saved, output similar to the following text is displayed:

   ```
   Successfully downloaded network localities.
   ```

3. Update the CSV file to specify changes to the network localities.

   For more information, see Consolidating network localities and Renaming network localities.
4. Run the `create_network_localities.py` script.
   - For sensors and ECA VMs, run the following command:

     ```
     python3 create_network_localities.py HOST --apikey API_KEY --group
       description
     ```

     Replace the following variables in the command with information from your ExtraHop system:

     • **HOST:** The IP address or hostname of the sensor or console.

- **API_KEY:** The API key.
- For Reveal(x) 360, run the following command:

```
python3 retrieve_network_localities.py HOST --id ID --secret SECRET --
group description
```

Replace the following variables in the command with information from your ExtraHop system:

- **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 /oauth/token.

- **ID:** The ID of the Reveal(x) 360 REST API credentials.

- **SECRET:** The secret of the Reveal(x) 360 REST API credentials.

The script adds each entry to the sensor or console. After each entry is added, output similar to the following text is displayed:

```
Successfully uploaded entry [auto]: Internal - 192.168.1.0/24
```