Back up a sensor or console through the REST API

Published: 2024-02-23

After you have configured a sensor or console with customizations such as bundles, triggers, and dashboards or administrative changes such as adding new users, ExtraHop recommends that you periodically create system backups to make it easier to recover from a system failure. This guide explains how to automate system backups through the ExtraHop REST API with a Python script.

Before you begin

- You must log in to the sensor or console with an account that has system and access administration 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 the ExtraHop REST API Guide ⤴ to learn how to navigate the ExtraHop REST API Explorer.

Retrieve and run the example Python script

The ExtraHop GitHub repository contains an example Python script that creates a backup file on a sensor or console and then downloads the backup file to the local machine. To automate system back ups, you can run this script periodically through a job scheduling application, such as cron.

> **Note:** Because you cannot download a file through the REST API Explorer, you must create backups through the REST API with a command or script.

1. Go to the ExtraHop code-examples GitHub repository ⤴ and download the `create_backup/create_backup.py` file to your local machine.
2. In a text editor, open the `create_backup.py` file and replace the following configuration variables with information from your environment:

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

   - **API_KEY:** The API key.

   - **BACKUP_NAME:** The name of the backup. The current timestamp is appended to this name when a backup is created. For example, if `BACKUP_NAME` is set to `Test`, a backup created on May 4th might be named `Test 2020-05-04 12-51-46.643813`.

3. Run the following command:

```
python create_backup.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)
> ```