Back up the ExtraHop system through the REST API

Published: 2020-08-11

After you have configured your ExtraHop system 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 ExtraHop system with an account that has 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 the ExtraHop REST API Guide to learn how to navigate the ExtraHop REST API Explorer.

Python script example

The following Python script creates a backup on an appliance and then downloads the backup file to the local machine. To automate system backups, 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.

The script includes the following configuration variables that you must replace with information from your environment:

• **HOST:** The IP address or hostname of the Discover or Command appliance.

• **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`.

```
#!/usr/bin/python3

import json
import requests
import sys

HOST = 'https://extrahop.example.com'
API_KEY = '123456789abcdefghijklmnop'
BACKUP_NAME = 'mybackup'

def createBackup(BACKUP_NAME):
    url = HOST + '/api/v1/customizations'
    headers = {
        'Authorization': 'ExtraHop apikey=%s' % API_KEY,
        'Content-Type': 'application/json'
    }
    data = {
        "name": BACKUP_NAME
    }
    r = requests.post(url, headers=headers, data=json.dumps(data))
    if r.status_code == 201:
```

```
        return True
    else:
        print('Unable to create backup')
        print(r.text)
        print(r.status_code)
        sys.exit()

def getIdName(BACKUP_NAME):
    url = HOST + '/api/v1/customizations'
    headers = {
        'Authorization': 'ExtraHop apikey=%s' % API_KEY,
        'Content-Type': 'application/json'
    }
    r = requests.get(url, headers=headers)
    if r.status_code == 200:
        backups = json.loads(r.text)
        for b in reversed(backups):
            if BACKUP_NAME in b['name']:
                return b['id'], b['name']
            else:
                continue
        print('Unable to retrieve ID for specified backup')
        sys.exit()
    else:
        print('Unable to retrieve backup IDs')
        print(r.text)
        print(r.status_code)
        sys.exit()

def downloadBackup(backup_id):
    url = HOST + '/api/v1/customizations/' + str(backup_id) + '/download'
    headers = {
        'Authorization': 'ExtraHop apikey=%s' % API_KEY,
        'accept': 'application/exbk'
    }
    r = requests.post(url, headers=headers)
    if r.status_code == 200:
        return r.content
    else:
        print('Unable to download backup')
        print(r.status_code)
        print(r.text)
        sys.exit()

def writeBackup(backup, BACKUP_NAME):
    new_name = BACKUP_NAME.replace(':', '')
    filepath = new_name + '.exbk'
    with open(filepath, "wb") as b:
        b.write(bytes(backup))
    print('Success! Backup file name:')
    print(filepath)

createBackup(BACKUP_NAME)
backup_id, BACKUP_NAME = getIdName(BACKUP_NAME)
backup = downloadBackup(backup_id)
writeBackup(backup, BACKUP_NAME)
```

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 ExtraHop system. Alternatively, you can add the `verify=False` option to bypass certificate verification. However, this method

is not secure and is not recommended. The following code sends an HTTP GET request without certificate verification:

```
requests.get(url, headers=headers, verify=False)
```