Migrate to SAML from LDAP through the REST API

Published: 2022-05-20

Secure, single sign-on (SSO) authentication to the ExtraHop system is easy to configure. However, if you have configured your ExtraHop system for remote authentication through LDAP, TACACS+, or RADIUS, changing to SAML permanently deletes all existing remote users and their customizations, such as saved dashboards, activity maps, reports, and record queries.

The ExtraHop GitHub repository provides a series of example scripts that show you how to safely migrate user customizations from remote users to SAML through the REST API. For each script, you must replace the script variables with information about your environment.

Important: Customizations must be saved from the ExtraHop system where remote users have created them. For example, if a remote user has a critical dashboard on a Command appliance and a Discover appliance, you must complete these procedures on both appliances for that remote user.

If you prefer to engage a turn-key solution for migration, contact your ExtraHop sales representative.

Procedure overview

Migrating to a new remote authentication method is a complex process. Be sure you understand all of the steps before you begin and be sure to schedule a maintenance window to avoid disrupting users.

Before you begin

1. Enable exception files on your ExtraHop systems. If the ExtraHop system unexpectedly stops or restarts during the migration process, the exception file is written to disk. The exception file can help ExtraHop Support diagnose the issue that caused the failure.
2. Create a backup of your ExtraHop systems. Backup files include all users, customizations, and shared settings. Download and store the backup file to a local machine.

Because changing the remote authentication method on the system effectively deletes all remote users, you must create SAML users on the system before you delete remote users. You can then transfer customizations owned by remote users to the SAML users as you delete the remote users.

Here is an explanation of each step:

1. Retrieve sharing metadata for customizations created by remote users.
2. (Optional for systems with a configured recordstore) Save record queries created by remote users to the setup user account.
3. Retrieve remote users and user groups.
4. Configure SAML on the system. (All remote users and user groups are deleted.)
5. Create SAML user accounts for each remote user that was deleted. After the system is configured for SAML, you can create a remote account for your users before they log in to the ExtraHop system for the first time.
6. Recreate local user groups that were deleted.
7. Delete remote user accounts and transfer customization sharing settings from the remote user accounts to the new SAML user accounts. When your SAML users log in for the first time, their customizations will be available.

Retrieve sharing metadata for remote user customizations

The ExtraHop GitHub repository contains an example script that retrieves a list of remote user customizations and associated sharing metadata and saves the information in JSON files. Run the script once for each type of customization after replacing the variables with information from your environment.

1. Go to the ExtraHop code-examples GitHub repository and download the `migrate_saml` directory to your local machine.
2. Set the following environment variables:

   EXTRAHOP_HOST

   The IP address or hostname of the ExtraHop system.

   EXTRAHOP_API_KEY

   The API key generated from the ExtraHop system.

   For example, the following Linux command sets the `EXTRAHOP_HOST` variable to `https://extrahop.example.com`:

   ```
   export EXTRAHOP_HOST=https://extrahop.example.com
   ```

3. Complete the following steps for both dashboards and activity maps.
   a) In a text editor, open the `retrieve_sharing.py` file and configure the following variables to specify the customization type. For example, to retrieve dashboard metadata, specify `OBJECT_TYPE=dashboards` and `OBJECT_FILE=dashboards.json`

      OBJECT_TYPE

      The type of customization metadata to retrieve. The following values are valid:

      - `dashboards`
      - `activitymaps`

      OUTPUT_FILE

      The name of the JSON file to save customization metadata in. Keep these files on your machine to input into scripts later in the migration.

      - `dashboards.json`
      - `activity_maps.json`

   b) Run the following command:

      ```
      python3 retrieve_sharing.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 ExtraHop system. 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)
      ```

Save record queries

In the following steps, you will learn how to preserve record queries saved by a remote user.

Because saved queries can be accessed by all system users, you can export all saved queries to a bundle and then upload them after migrating to SAML. Imported record queries are assigned to the user that uploads the bundle. (For example, if you import queries from a bundle while logged in as the setup user, all of the

queries list setup as the query owner.) After migration, remote users can view the saved record queries and save a copy for themselves.

1. Log in to the ExtraHop system through `https://<extrahop-hostname-or-IP-address>` with the `setup` user account.
2. Click the System Settings icon and then select **Bundles**.
3. From the Bundles page, select **New**.
4. Type a name to identify the bundle.
5. Click the arrow next to Queries in the Contents table and select the checkboxes next to the saved queries you want to export.
6. Click **OK**. The bundle appears in the table on the Bundles page.
7. Select the bundle and click **Download**. The queries are saved to a JSON file.

Next steps
After migration, upload the bundle to restore the saved record queries.

Retrieve remote users

The ExtraHop GitHub repository contains an example script that retrieves a list of remote users and their associated metadata and then saves the information in a JSON file named `user_map.json`.

In the `migrate_saml` directory downloaded from the ExtraHop code-examples GitHub repository, run the following command:

```
python3 retrieve_remote_users.py
```

Important: If an ExtraHop system includes duplicate LDAP user account names, the script will fail and list the duplicate names in the output. LDAP user account names are case sensitive, but SAML user account names are not. You must rename duplicate LDAP user account names before migrating them. For example, if you have LDAP user names `user_1` and `User_1`, you must rename one of those accounts before migrating to SAML.

Retrieve local user groups

The ExtraHop GitHub repository contains an example script that retrieves a list of local user groups and members and then saves the information in a JSON file named `user_groups.json`.

In the `migrate_saml` directory downloaded from the ExtraHop code-examples GitHub repository, run the following command:

```
python3 retrieve_local_user_groups.py
```

Configure SAML on the ExtraHop system

Depending on your environment, configure SAML. Guides are available for both Okta and Google. After you configure SAML on your ExtraHop system, you are able to create accounts on the ExtraHop system for your remote users, and transfer their customizations before they log in for the first time.

Create SAML user accounts

The ExtraHop GitHub repository contains an example script that creates SAML user accounts for each deleted remote user account on an ExtraHop system.

Note: Verify the required format for usernames that are entered in the Login ID field with the administrator of your Identity Provider. If the usernames do not match, the remote user will not be matched to the user created on the ExtraHop system.

Note: The script generates SAML usernames through the `generateName()` method. By default, the script creates new usernames by appending `@example.com` to the end of the remote username. You must configure the method to generate usernames according to your SAML user account naming standard. Verify how to format usernames with the administrator of your Identity Provider.

You can also specify SAML usernames in a CSV file. To configure the script to retrieve usernames from a CSV file, set the `READ_CSV_FILE` variable in the script to `True`. The CSV file must meet the following requirements:

- The CSV file must not contain a header row.
- Each row of the CSV file must contain the following two columns in the specified order:

ExtraHop username	SAML username

- The CSV file must be named `remote_to_saml.csv` and be located in the same directory as the Python script. The `migrate_saml` directory contains an example CSV file named `remote_to_saml.csv`.

In the `migrate_saml` directory downloaded from the ExtraHop code-examples GitHub repository, run the following command:

```
python3 create_saml_accounts.py
```

Recreate local user groups

The ExtraHop GitHub repository contains an example script that restores membership for SAML users to local user groups.

In the `migrate_saml` directory downloaded from the ExtraHop code-examples GitHub repository, run the following command:

```
python3 create_local_user_groups.py
```

Delete remote user accounts

The ExtraHop GitHub repository contains an example script that deletes remote user accounts and transfers the customizations owned by those user accounts to SAML user accounts.

In the `migrate_saml` directory downloaded from the ExtraHop code-examples GitHub repository, run the following command:

```
python3 delete_remote_users.py
```

Transfer customization sharing settings to SAML user accounts

The ExtraHop GitHub repository contains an example script that transfers customization sharing settings from deleted remote user accounts to SAML user accounts. Run the script once for each type of customization after replacing the variables with information from your environment. For example, if you want to preserve shared settings for dashboards and activity maps, you will run the script once with the customization variables for dashboards and once with the customization variables for activity maps.

In the `migrate_saml` directory downloaded from the ExtraHop code-examples GitHub repository, complete the following steps for dashboards, activity maps, and reports.

a) In a text editor, open the `transfer_sharing.py` file and configure the following variables to specify the customization type. For example, to retrieve dashboard metadata, specify `OBJECT_TYPE=dashboards` and `OBJECT_FILE=dashboards.json`

OBJECT_TYPE

The type of customization to transfer. The following values are valid:

- `dashboards`
- `activitymaps`
- `reports`

OBJECT_FILE

The name of the JSON file that includes the customization metadata. These files must be located in the same directory as the Python script along with the `user_map.json` file that contains the list of remote users and the `user_groups.json` file that contains the list of user groups. The following values are valid:

- `dashboards.json`
- `activity_maps.json`
- `reports.json`

b) Run the following command:

```
python3 transfer_sharing.py
```