Configure Microsoft Graph API Integration with Cisco XDR

Contents

Introduction

Prerequisites

Integration Steps

Perform Investigations

Verify

Troubleshoot

Introduction

This document describes the proceedure to integrate Microsoft Graph API with Cisco XDR, and the type of data that can be queried.

Prerequisites

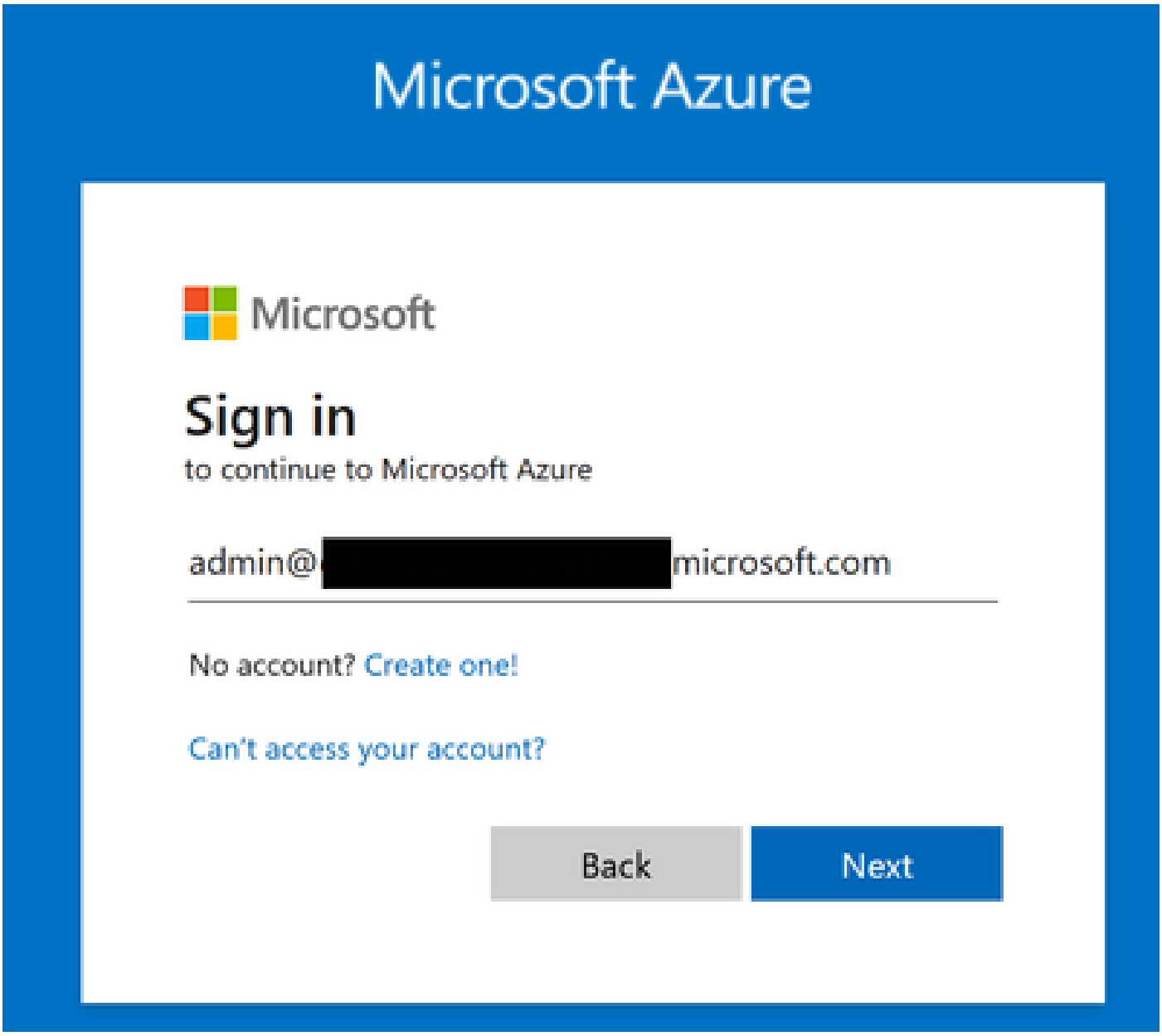
- Cisco XDR Admin Account
- Microsoft Azure System Administrator Account
- Access to Cisco XDR

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Integration Steps

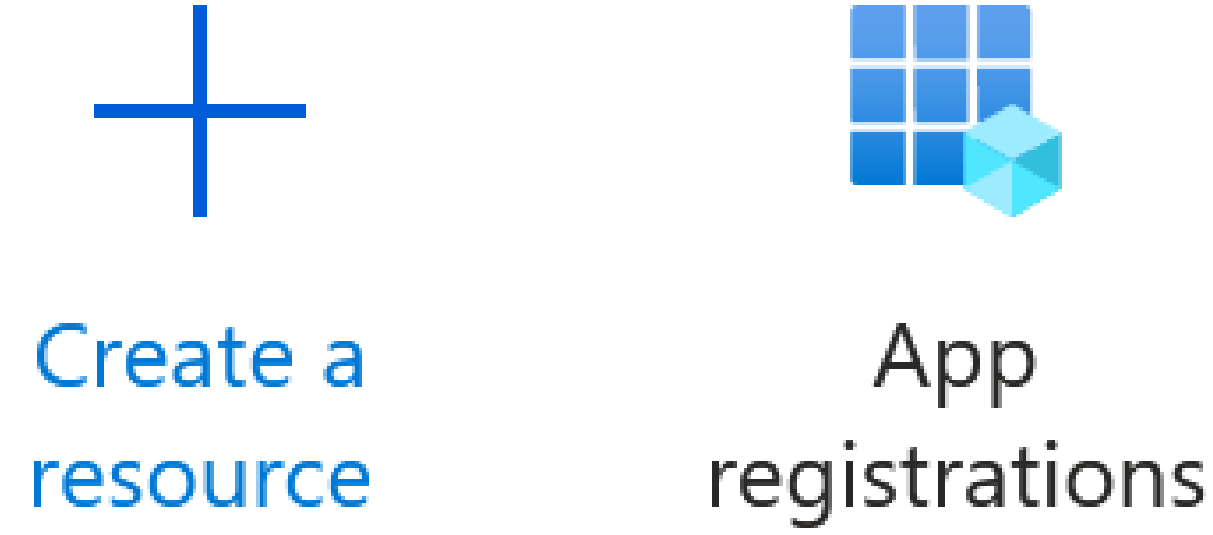
Step 1.

Log in into Microsoft Azure as a System Administrator.



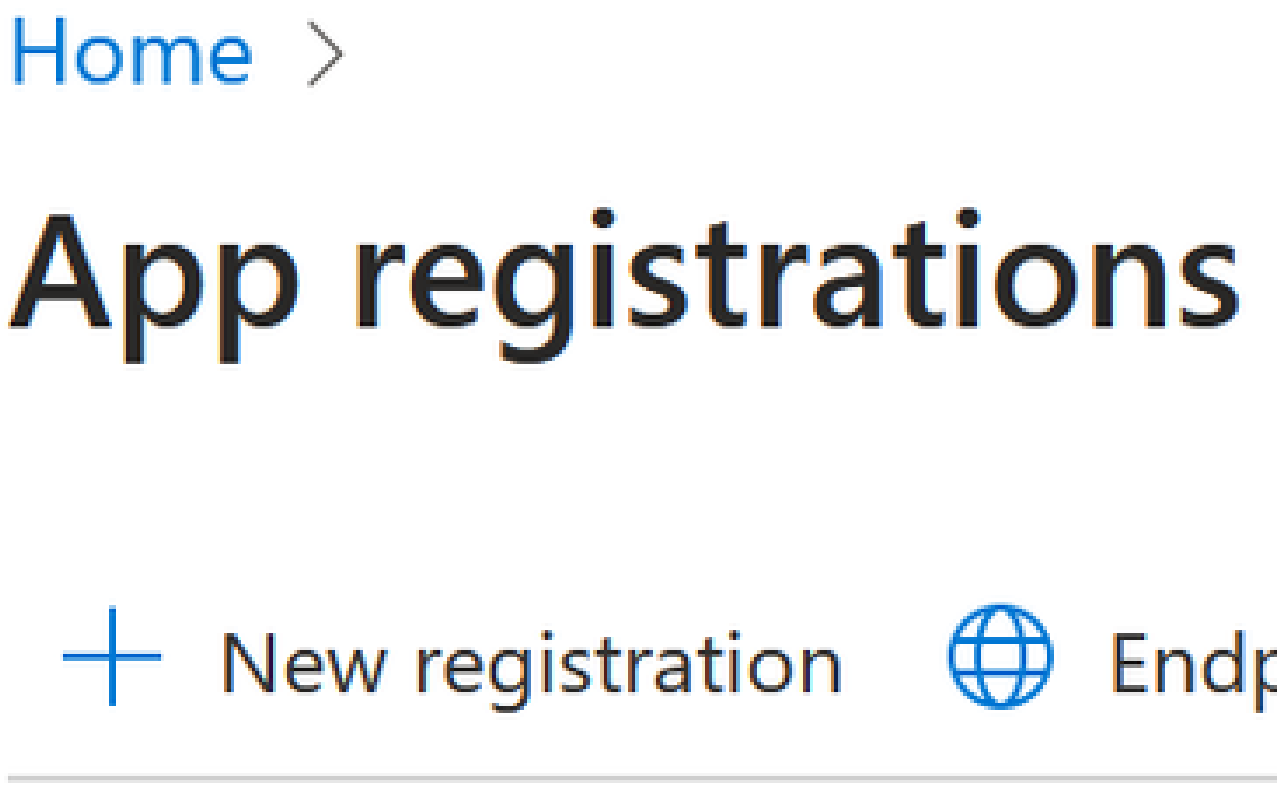
Step 2.

Click **App Registrations** on the Azure services portal.



Step 3.

Click New registration**.**



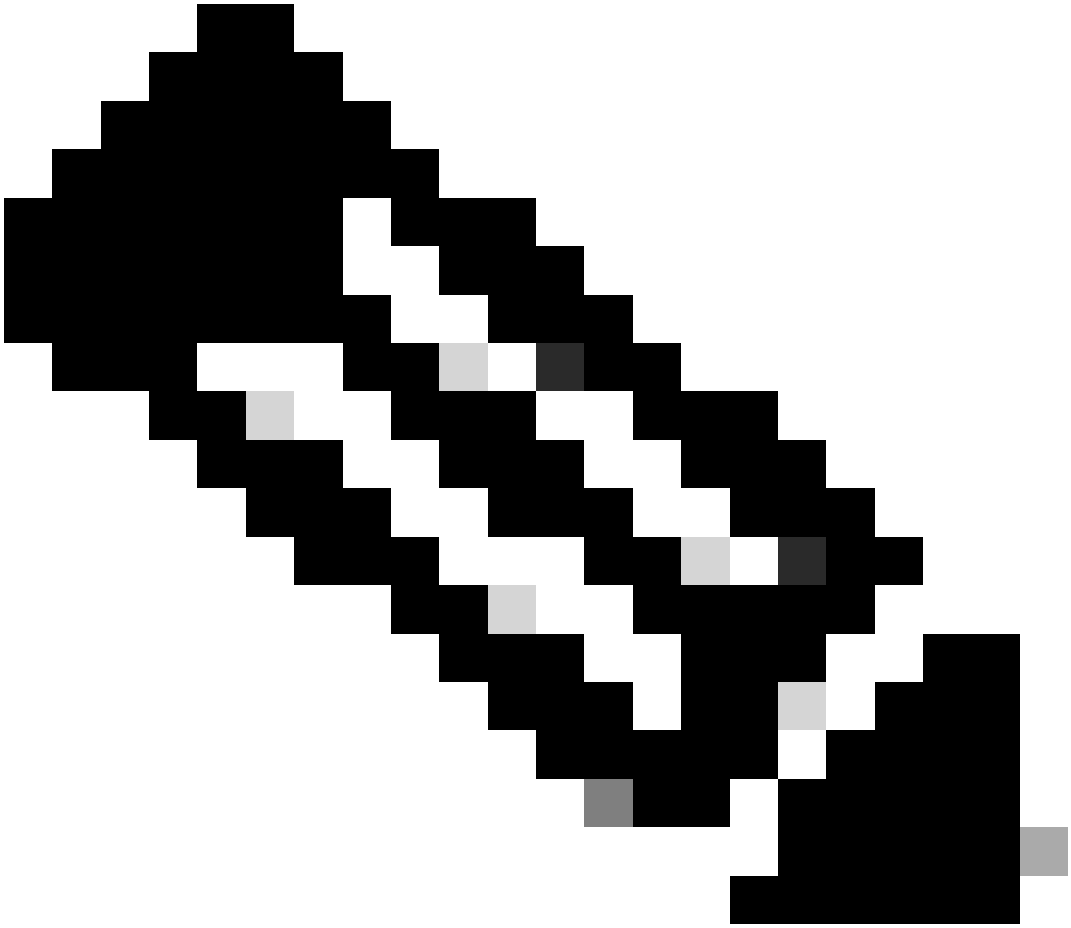
Step 4.

Type a name to identify your new App.

* Name

The user-facing display name for this application (this can be changed later).

SecureX - Graph API



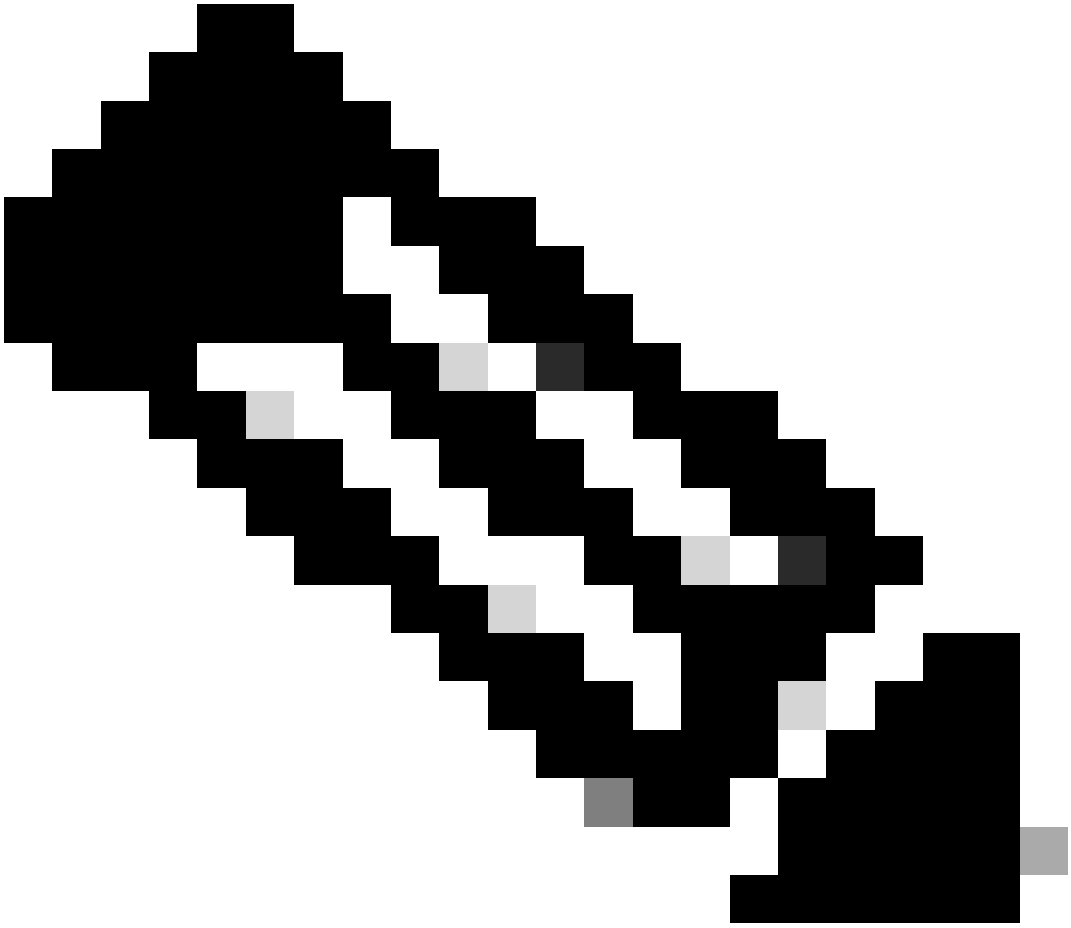
Note: A green check mark appears if the name is valid.

On supported Account Types, choose the option **Accounts in this organizational directory only.**

Supported account types

Who can use this application or access this API?

- (•) Accounts in this organizational directory only ([redacted] - Single tenant)
- ( ) Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- ( ) Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- ( ) Personal Microsoft accounts only



Note: You do not need to type a Redirect URI.

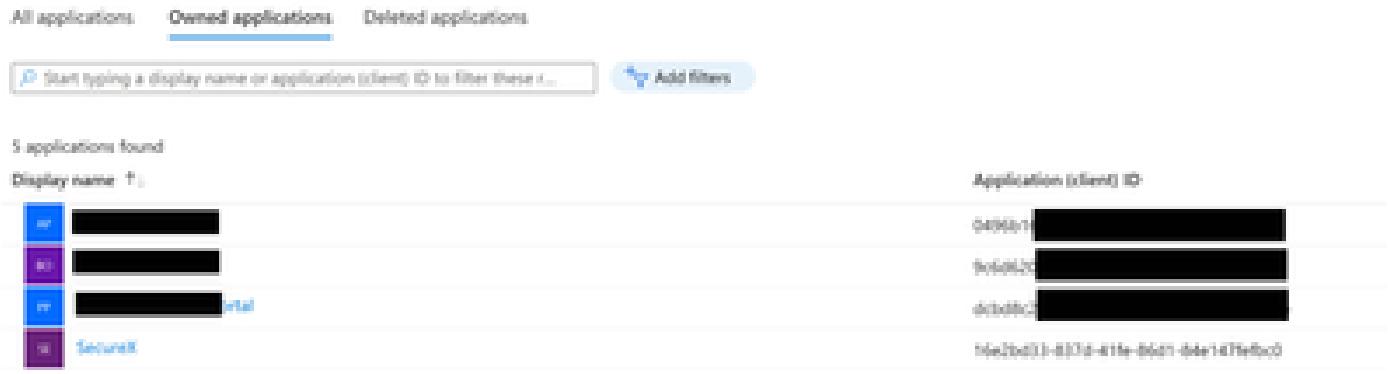
Step 5.

Scroll to the bottom of the screen and click **Register**.

Step 6.

Navigate back to the Azure services page, click App Registrations > Owned Applications.

Identify your App and click the name. In this example, it is SecureX.



Step 7.

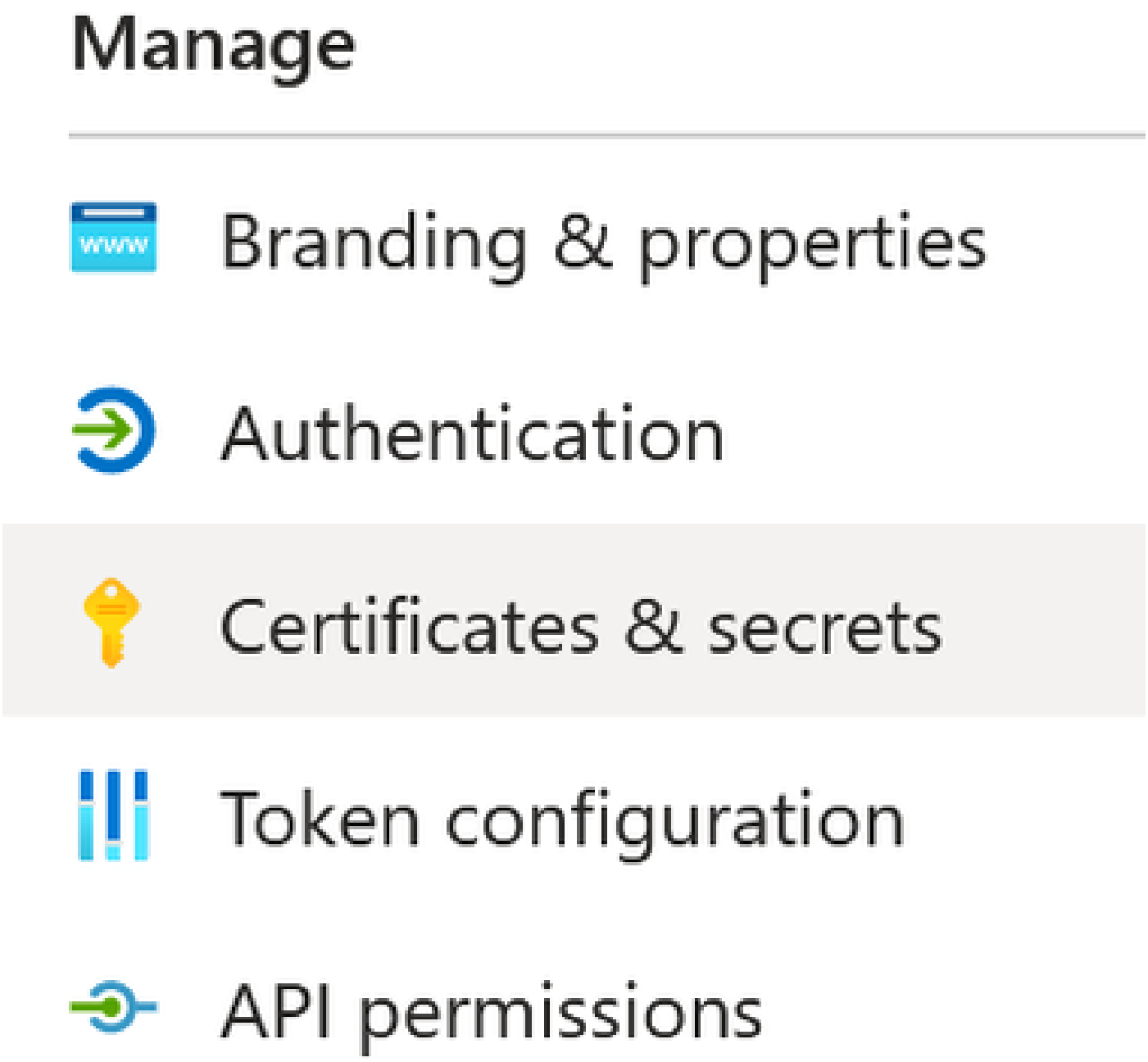
A summary of your App appears. Please identify these relevant details:

Application (client) ID:

Directory (tenant) ID:

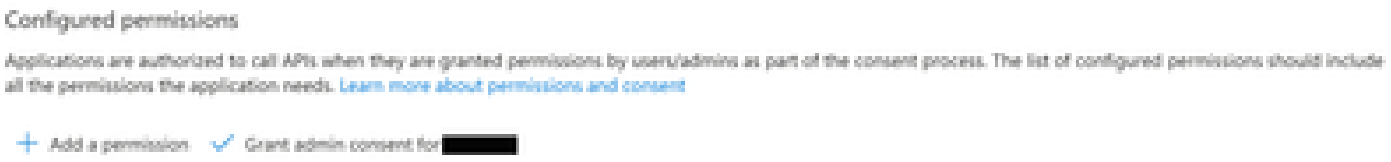
Step 8.

Navigate to Manage Menu > API Permissions.



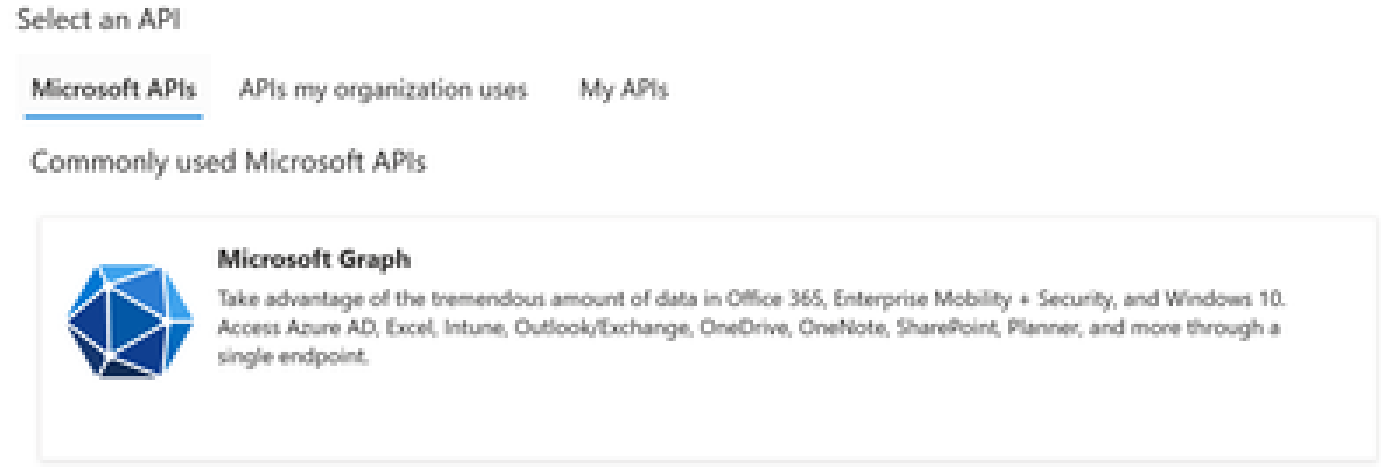
Step 9.

Under Configured Permissions, click Add a Permission.



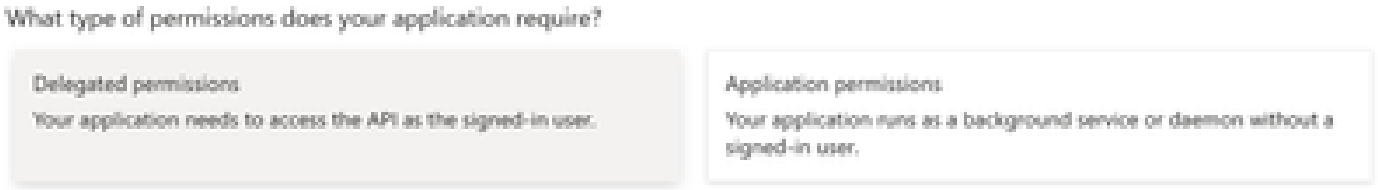
Step 10.

In the section Request API Permissions, click **Microsoft Graph.**



Step 11.

Select Application permissions.



In the Search bar, look for Security. Expand **Security Actions** and select

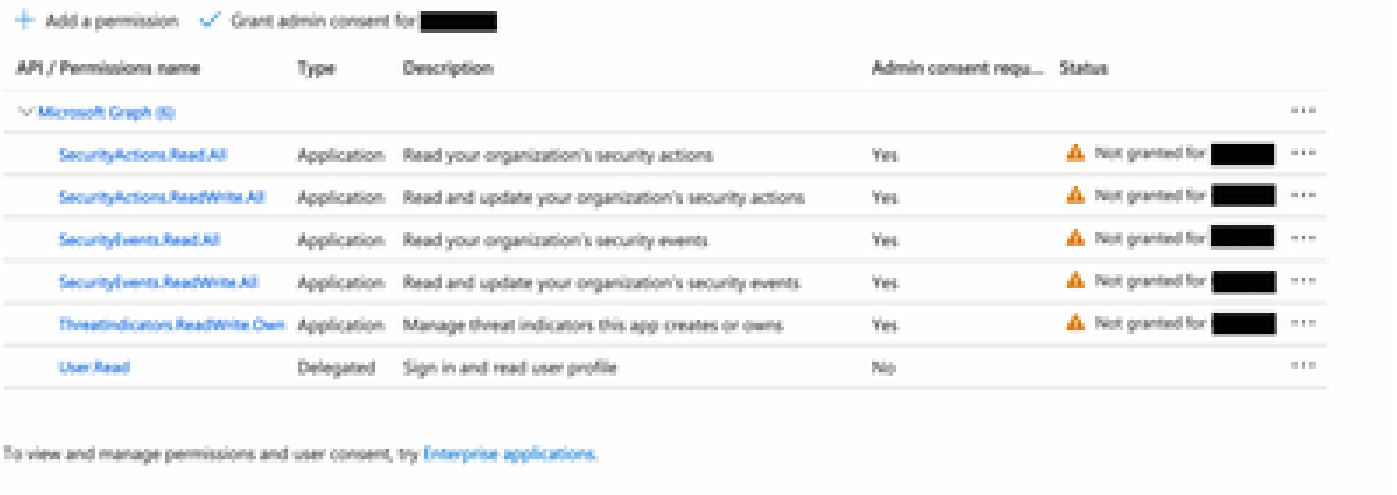
- **Read.All**
- **ReadWrite.All**

- **Security Events** and select
  - **Read.All**
  - **ReadWrite.All**
- **Threat indicators** and select
  - **ThreatIndicators.ReadWrite.OwnedBy**

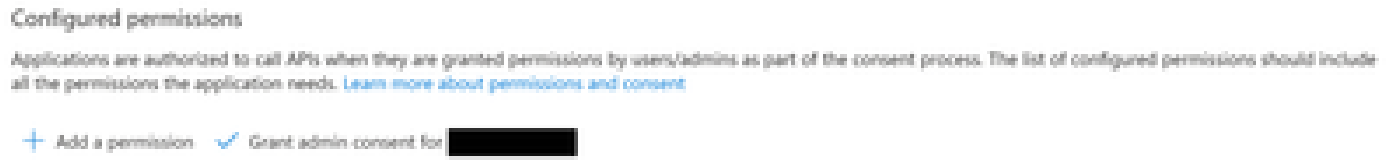
Click Add permissions.

Step 12.

Review your selected permissions.

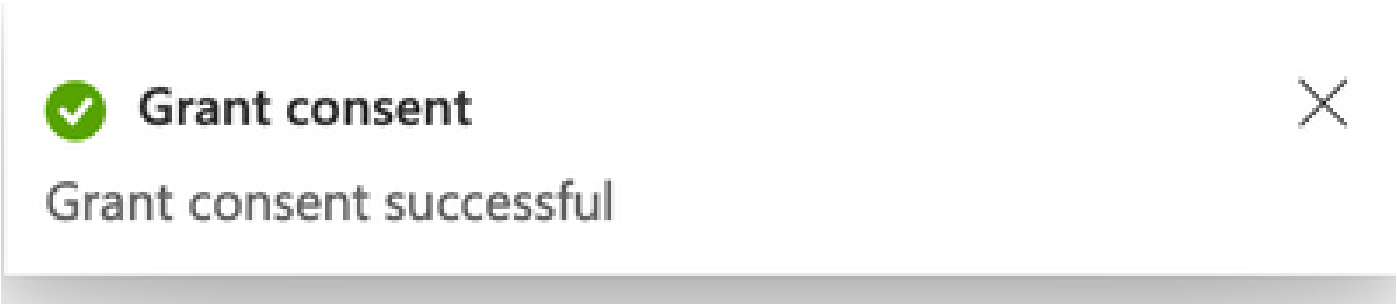


Click **Grant Admin consent** for your organization.



A prompt to choose if you want to grant consent for all the permissions appears. Click Yes.

A similar popup as shown in this image appears:



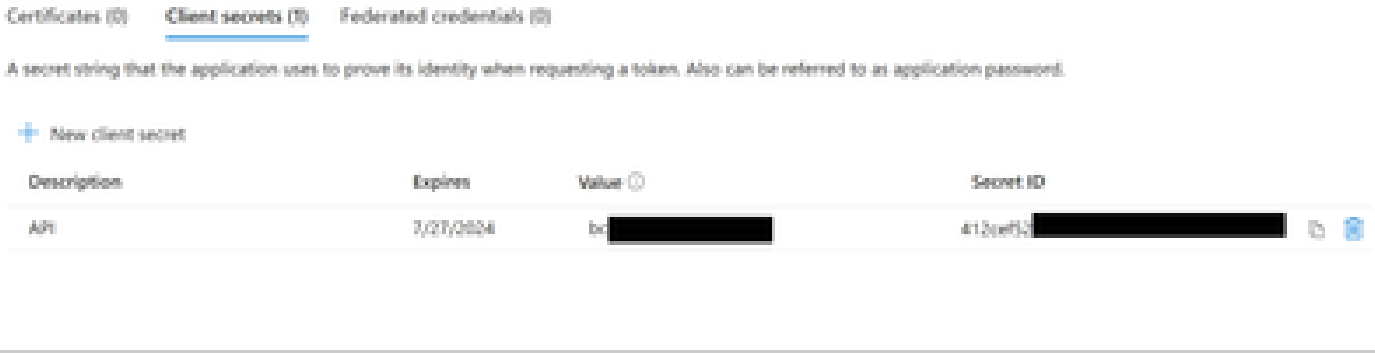
Step 13.

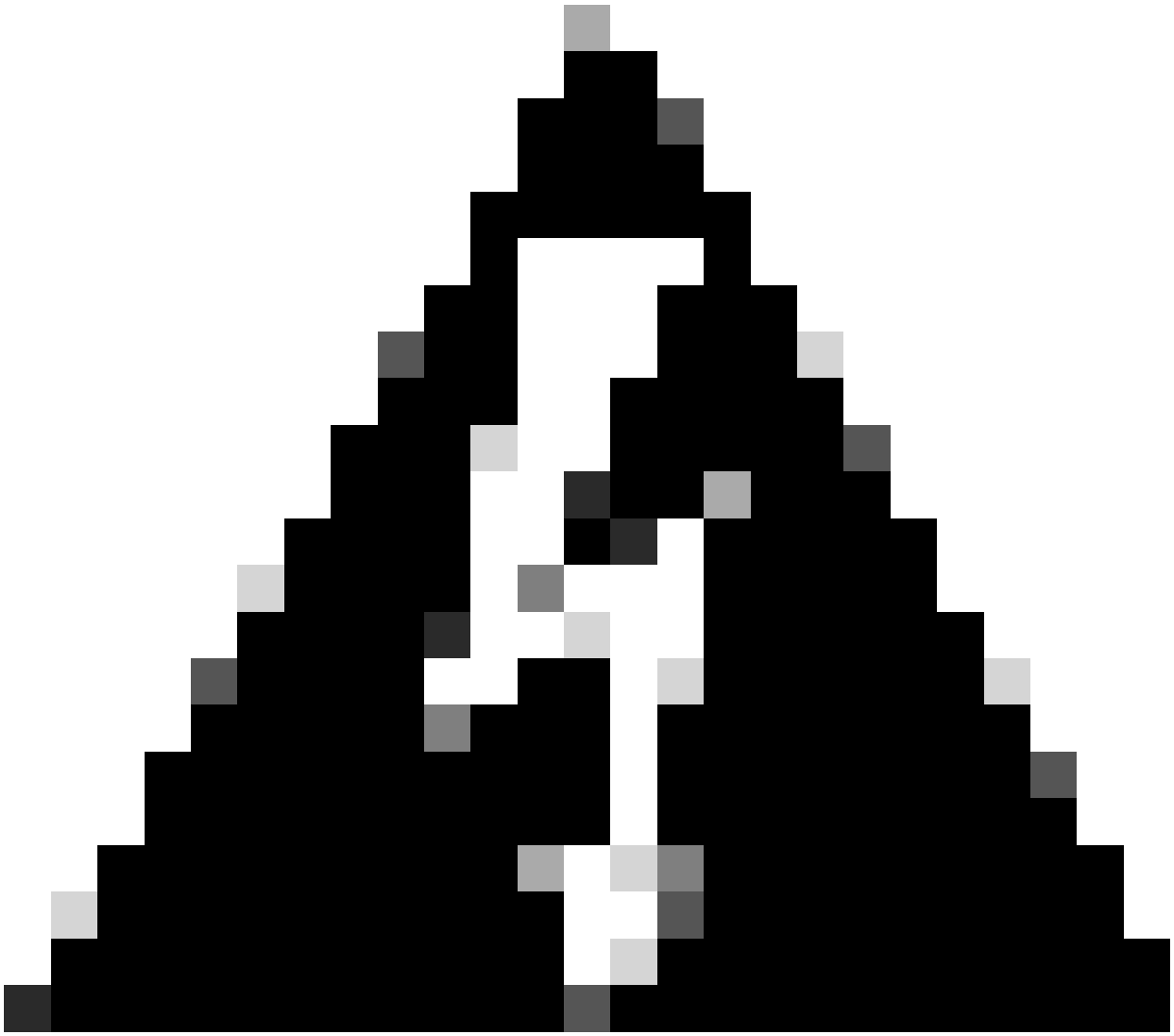
Navigate to Manage > Certificates & Secrets.

Click Add New Client Secret.

Write a brief description and select a valid Expires date. Iit is suggested to select a validity date of over 6 months to prevent the API keys expiration.

Once created, copy and store in a safe place the portion that says **Value**, as it is used for the integration.





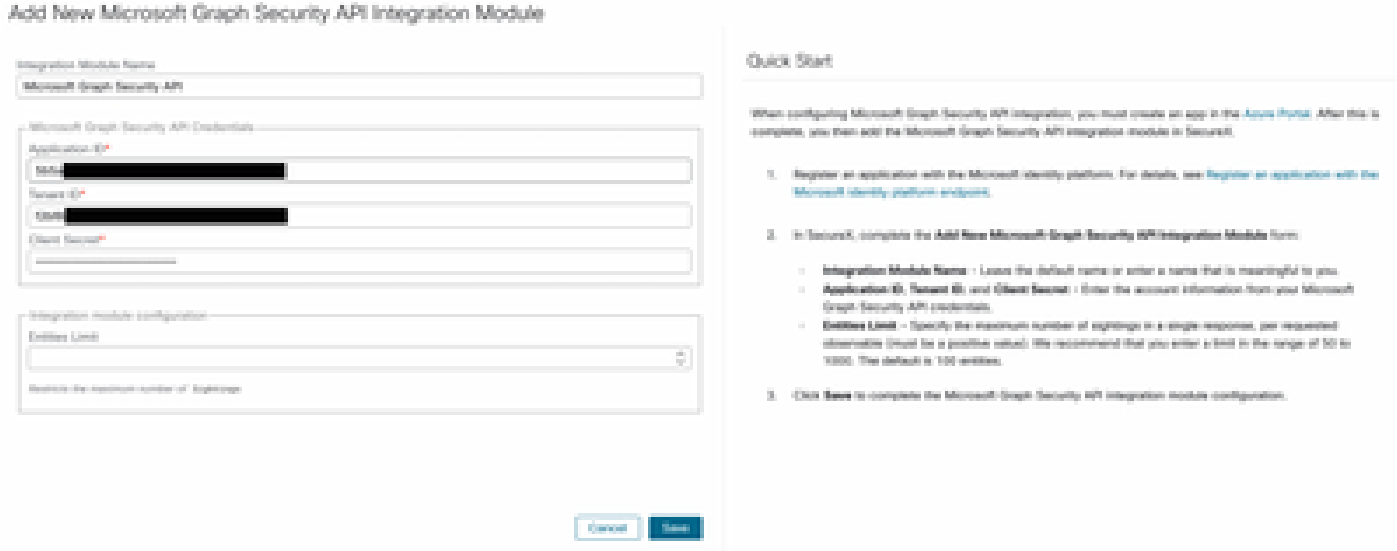
Warning: This field cannot be recovered and you must to create a New Secret.

Once you have all the information, navigate back to **Overview** and copy the values of your App. Then navigate to SecureX.

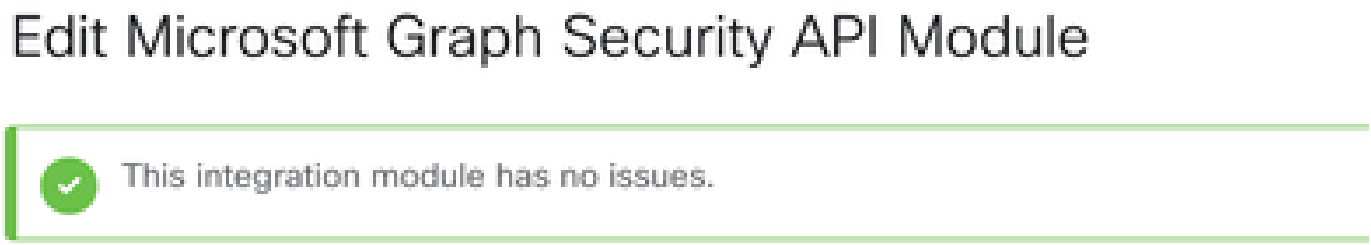
Step 14.

Navigate to Integration Modules > Available Integration Modules > select Microsoft Security Graph API, click Add**.**

Assign a name and paste the values you got from the Azure portal.



Click Save and wait for the Healtcheck to succeed.



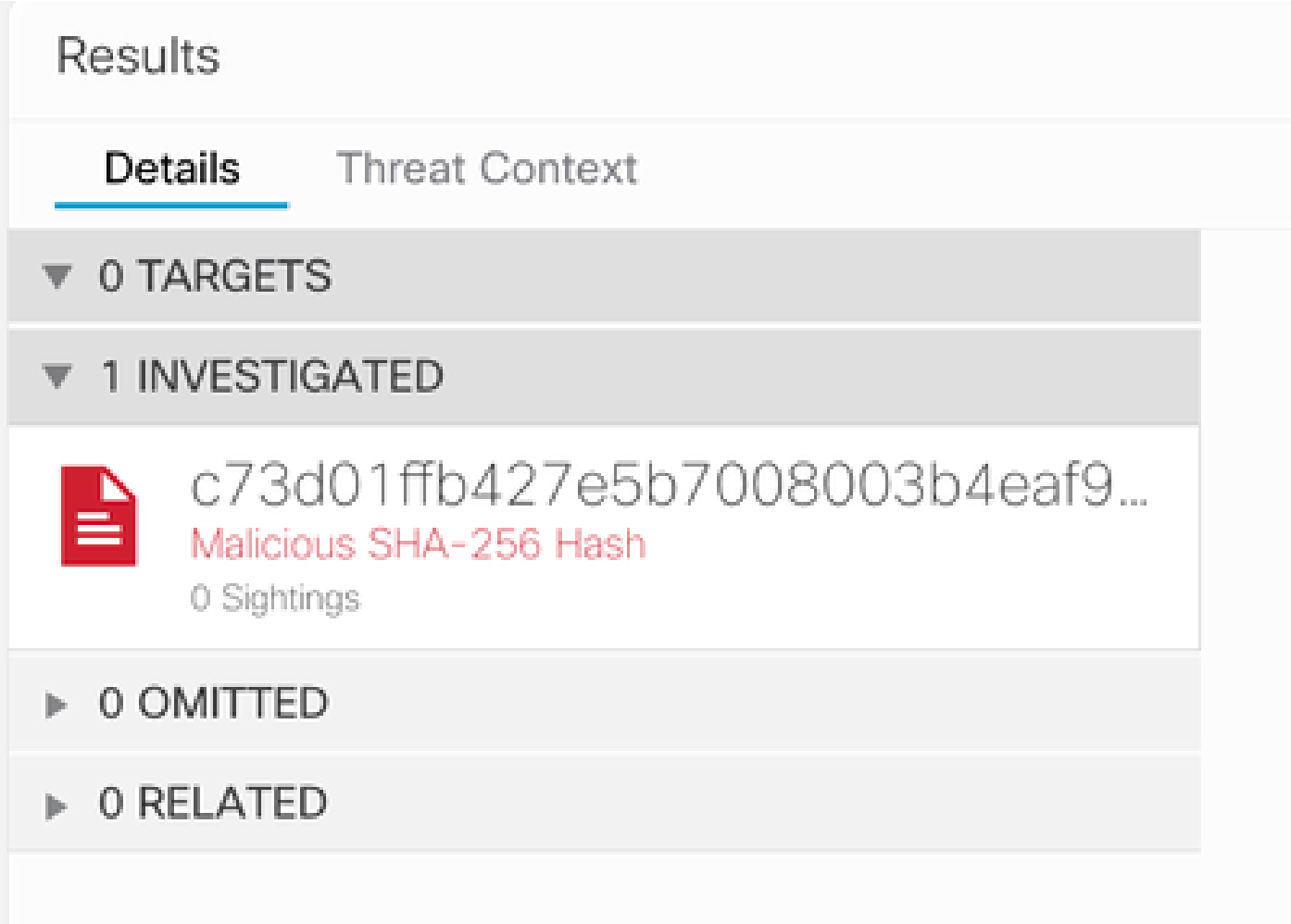
Perform Investigations

As of now, Microsoft Security Graph API does not populate the Cisco XDR Dashboard with a tile. Rather, the information from your Azure portal can be queried with the use of Investigations.

Keep in mind, the Graph API can only be queried for:

- ip
- domain
- hostname
- url
- file_name
- file_path
- sha256

In this example, the investigation used this SHA c73d01ffb427e5b7008003b4eaf9303c1febd883100bf81752ba71f41c701148.



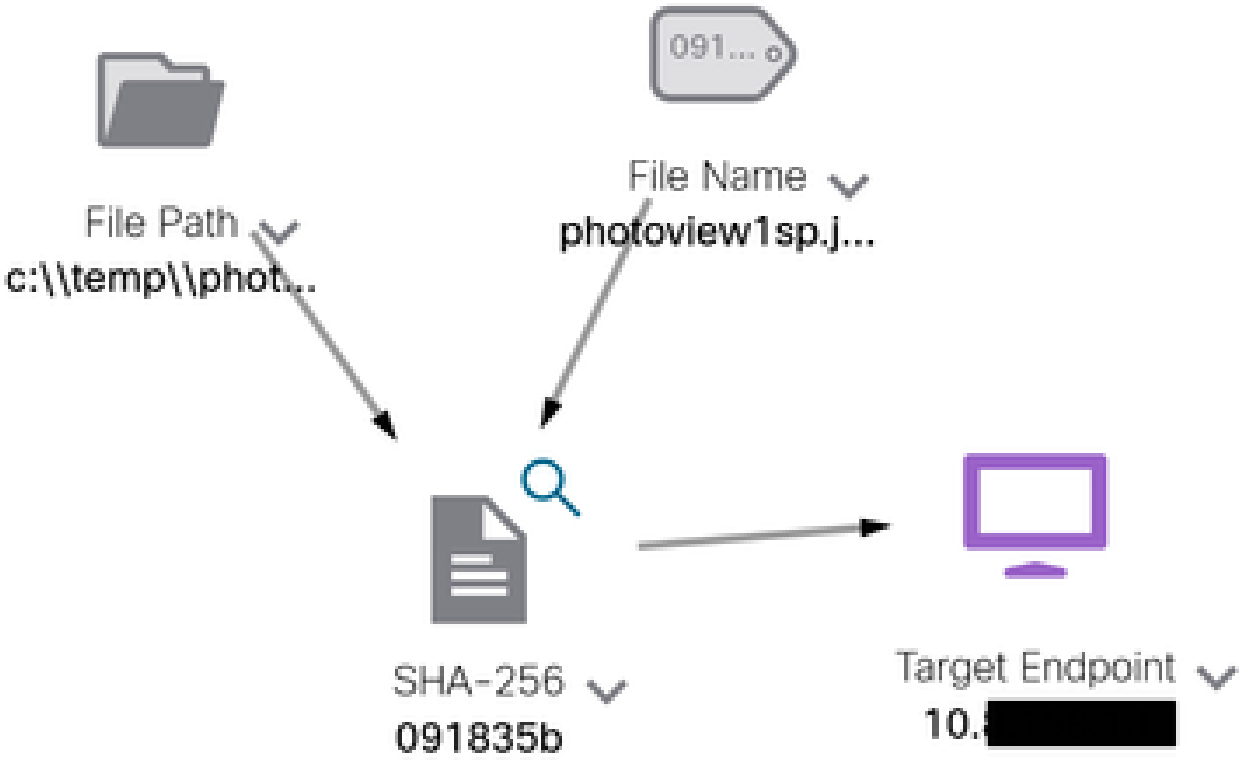
As you can see, it has 0 Sightings in the Lab Environment, so how to test if Graph API works?

Open the WebDeveloper Tools, run the investigation, find a Post Event to **visibility.amp.cisco.com** the file called Observables.

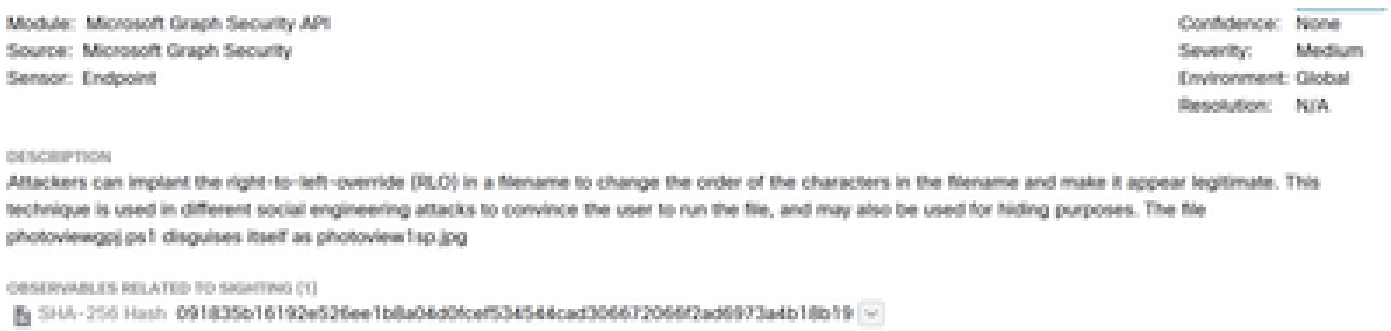
Verify

You can use this link: [Microsoft graph security Snapshots] for a list of Snapshots that help you understand the response you can get from each type of observable.

You can see an example as shown in this image:



Expand the window, you can see the information provided by the integration:



Keep in mind that data has to exist in your Azure portal, and Graph API works better when used with other Microsoft solutions. However, this has to be validated by Microsoft Support.

Troubleshoot

- Authorization Failed Message:
  - Ensure the values for **Tenant ID** and Client ID are correct and that they are still valid.

- No Data appears in Investigation:
  - Ensure you copied and pasted appropriate values for **Tenant ID** and **Client ID**.
  - Ensure you used the information of the field **Value** from the Certificates & Secrets section.
  - Use WebDeveloper tools to determine if the Graph API is queried when an investigation occurs.
  - As the Graph API merges data from various Microsoft alert providers, ensure that OData is supported for the query filters. (For example, Office 365 Security and Compliance and Microsoft Defender ATP).