

Configure Microsoft Graph API Integration with SecureX

Contents

[Introduction](#)

[Prerequisites](#)

[Integration Steps](#)

[Perform Investigations](#)

[Verify](#)

[Troubleshoot](#)

Introduction

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

Prerequisites

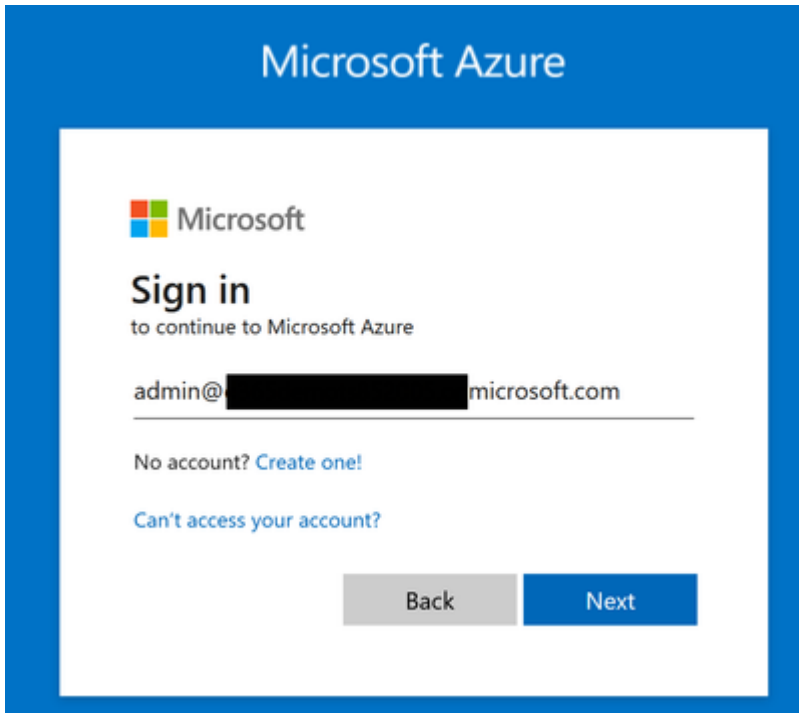
- SecureX Admin Account
- Microsoft Azure System Administrator Account
- Access to SecureX Threat Response

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.



Create a
resource



App
registrations

Step 3.

Click **New registration**.

[Home](#) >

App registrations

[+](#) New registration [🌐](#) Endp

Step 4.

Type a name to identify your new App.

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

 ✓

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 (██████████ 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**.

By proceeding, you agree to the [Microsoft Platform Policies](#)

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**.

All applications **Owned applications** Deleted applications

Start typing a display name or application (client) ID to filter these ... Add filters

5 applications found

Display name	Application (client) ID
██████████	0496517-██████████
██████████	7656622-██████████
██████████	d43d32c-██████████
SecureX	16a2b033-837d-419e-86d1-04e1479efc03

Step 7.

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

Application (client) ID:

Display name : [SecureX](#)
Application (client) ID : 16e2bd33- [REDACTED]

Directory (tenant) ID:

Directory (tenant) ID : f2bf8cd3 [REDACTED]


Step 8.

Navigate to Manage Menu > API Permissions.

Manage

 Branding & properties

 Authentication

 Certificates & secrets

 Token configuration

 API permissions

Step 9.

Under Configured Permissions, click Add a Permission.

Configured permissions
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)
+ Add a permission ✓ Grant admin consent for [REDACTED]


Step 10.

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

Select an API

Microsoft APIs APIs my organization uses My APIs

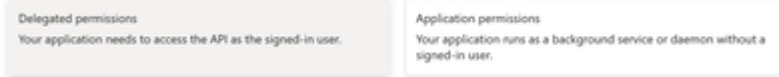
Commonly used Microsoft APIs

**Microsoft Graph**
Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.

Step 11.

Select Application permissions.

What type of permissions does your application require?



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.

API / Permissions name	Type	Description	Admin consent req...	Status
SecurityActions.Read.All	Application	Read your organization's security actions	Yes	Not granted for [redacted] ...
SecurityActions.ReadWrite.All	Application	Read and update your organization's security actions	Yes	Not granted for [redacted] ...
SecurityEvents.Read.All	Application	Read your organization's security events	Yes	Not granted for [redacted] ...
SecurityEvents.ReadWrite.All	Application	Read and update your organization's security events	Yes	Not granted for [redacted] ...
ThreatIndicators.ReadWrite.Own	Application	Manage threat indicators this app creates or owns	Yes	Not granted for [redacted] ...
User.Read	Delegated	Sign in and read user profile	No	...

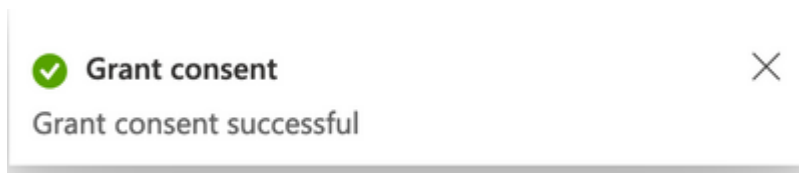
To view and manage permissions and user consent, try [Enterprise applications](#).

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. It 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.

Certificates (0) **Client secrets (1)** Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	Secret ID
API	7/27/2024	br [REDACTED]	412ceff5 [REDACTED]

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.



Microsoft Graph Security API

The Microsoft Graph Security API is an intermediary service that provides a single programmatic interface to connect multiple Microsoft Graph Security providers. Requests to the...

[+ Add](#) [Learn More](#)

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

Add New Microsoft Graph Security API Integration Module

Integration Module Name: Microsoft Graph Security API

Microsoft Graph Security API Credentials

Application ID: [REDACTED]

Secret ID: [REDACTED]

Client Secret: [REDACTED]

Integration Module Configuration

Entities Limit: [REDACTED]


Quick Start

When configuring Microsoft Graph Security API integration, you must create an app in the Azure Portal. After this is complete, you then add the Microsoft Graph Security API integration module in SecureX.

1. Register an application with the Microsoft identity platform. For details, see [Register an application with the Microsoft identity platform endpoint](#).
2. In SecureX, complete the [Add New Microsoft Graph Security API Integration Module Form](#).
 - **Integration Module Name** - Leave the default name or enter a name that is meaningful to you.
 - **Application ID**, **Secret ID**, and **Client Secret** - Enter the account information from your Microsoft Graph Security API credentials.
 - **Entities Limit** - Specify the maximum number of entities in a single response, per requested identifier (must be a positive integer). We recommend that you enter a limit in the range of 50 to 1000. The default is 100 entities.
3. Click **Save** to complete the Microsoft Graph Security API integration module configuration.

Click save and wait for the Healthcheck to succeed.

Edit Microsoft Graph Security API Module

 This integration module has no issues.

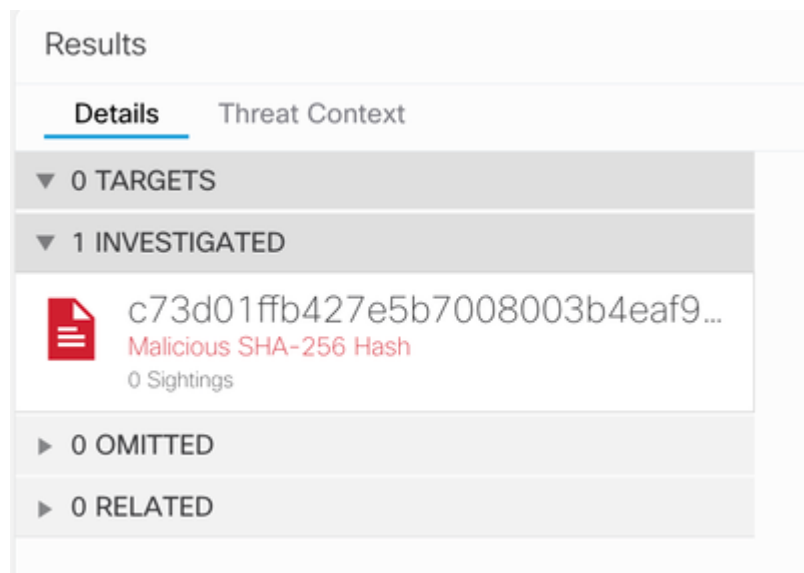
Perform Investigations

As of now, Microsoft Security Graph API does not populate the SecureX 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?

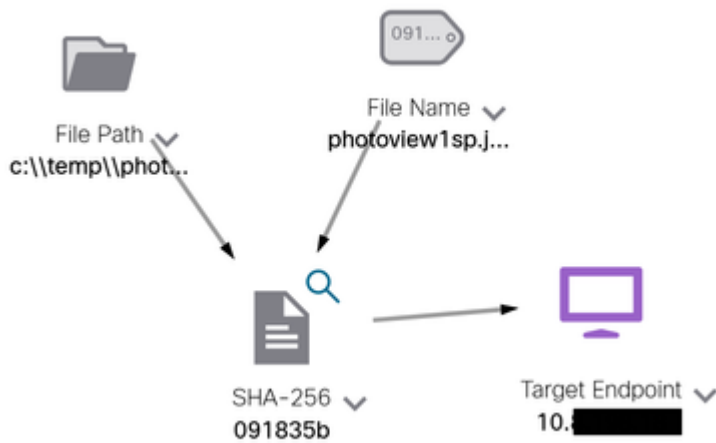
Open the WebDeveloper Tools, run the investigation, find a Post Event to **visibility.amp.cisco.com** the file called Observables.



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:

Module: Microsoft Graph Security API
 Source: Microsoft Graph Security
 Sensor: Endpoint

Confidence: None
 Severity: Medium
 Environment: Global
 Resolution: N/A

DESCRIPTION
 Attackers can implant the right-to-left-override (RLO) in a filename to change the order of the characters in the filename and make it appear legitimate. This technique is used in different social engineering attacks to convince the user to run the file, and may also be used for hiding purposes. The file photoviewggj.pp1 disguises itself as photoview1sp.jpg

OBSERVABLES RELATED TO SIGHTING (1)
 SHA-256 Hash: 091835b16192e526ee1b8a04d0fce534544cad306672066f2ad6973a4b18b19

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.

For reference on TAC Scope, please refer to this document: [Verify the support scope for securex.](#)

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).