MICROSOFT SENTINEL AND DEFENDER ATP INTEGRATIONS ADMIN GUIDE (AZURE LOGIC APP VERSION)

This integration is deprecated. Use the **Microsoft Sentinel and Defender ATP Integrations Admin Guide (Docker Version)** (https://docs.mandiant.com/home/mati-microsoft-sentinel-and-defender-integrations-docker-setup-guide).

Overview

The Mandiant integration with Microsoft Sentinel and Defender ATP uses the Microsoft Azure Logic App framework to collect indicators of compromise from Mandiant and ingest them into either Microsoft Sentinel or Defender ATP. This integration brings Mandiant Advantage Threat Intelligence (MATI) to Microsoft Sentinel and Defender ATP, highlighting indicators of compromise in your network and allowing you to identify and explore those that matter most.

A Docker-based version of this integration has been released as an alternative solution that addresses supportability and reliability issues with the Microsoft Azure Logic App framework. See **Microsoft Sentinel and Defender ATP Integrations Admin Guide (Docker Version)** (https://docs.mandiant.com/home/mati-microsoft-sentinel-and-defender-integrations-docker-setup-guide).

Prerequisites

In order to use this integration, you need:

- A valid Mandiant Threat Intelligence Subscription
- A valid Microsoft Azure subscription
- An instance of Microsoft Sentinel and/or Microsoft Defender ATP
- A Mandiant Threat Intelligence API public and secret key
- A Microsoft Azure Storage Account and the Storage Account Access Key
- A Microsoft Azure administrator account with permissions to
  - Deploy a Logic App
  - Authorize a connection to the Microsoft Graph Security API
  - Create a Microsoft Storage Account: `Microsoft.Storage/storageAccounts/write`
  - Authorize a connection to Microsoft Azure Blob storage
  - Create a Microsoft Key Vault: `Microsoft.KeyVault/vaults/write`

Get API Key ID and Secret

To obtain a **Service API Key** (which is tied to an organization rather than an individual user) for use with third-party security technologies such as a SIEM, contact **Support** (https://www.mandiant.com/support).

To obtain an API Key ID and Secret for an individual user account, perform the following:

1. Navigate to the Mandiant Threat Intelligence web console.
2. Click **Account Settings**.
3. Select **API Access and Keys** from the navigation menu.
4. Click **Get Key ID and Secret**.
5. Copy and store the displayed values in a secure location.

Setup

The setup consists of the following steps:

1. **Deploy the Dependencies** (the Azure Key Vault and the Azure Blob Storage account).
2. **Enable the Sentinel Threat Intelligence Connector** if you are using Microsoft Sentinel.
3. **Deploy the Azure Logic App** (using the Default template).
4. **Authorize Connections**.
5. **Assign Access Policy**.
6. **Configure the Logic App Settings**.

Deploy the Dependencies

Click the button below to deploy the dependencies using the Dependency template. Your browser will be redirected to Microsoft Azure.

Deploy to Azure

(https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fwww%2Egstatic%2Ecom%2Fcloud%2Dmandiant%2Deng%2Dintegrations%2Fthreat%5Fintel%2Fazur

If you wish to create the resources manually, you may create the following resources (rather than having them deployed via the template), then follow all other remaining steps of the integration **setup**:

1. An Azure Key Vault
2. A Key, within the Key Vault, containing the Secret Key for the Mandiant Threat Intel API
3. An Azure Blob Storage account
4. A Container inside of the Azure Blob Storage account

Enable the Sentinel Threat Intelligence Connector

Only applicable when ingesting indicators for use in Microsoft Sentinel

1. While logged in to Microsoft Sentinel, open the **Data connectors** page from the **Configuration** menu.

2. Locate and click on the **Threat Intelligence Platforms (Preview)** connector.
3. Click the **Open connector page** button from the sliding panel on the right-hand side of the page.
4. Ensure you have the proper prerequisites to connect. You should be able to identify 2 green checkmarks:
    a. **Workspace**: read and write permissions.
    b. **Tenant Permissions**: 'Global Administrator' or 'Security Administrator' on the workspace's tenant.
5. Scroll to the bottom of the **Threat Intelligence Platforms (Preview)** page and click the **Connect** button.

The connector is now enabled.

Deploy the Azure Logic App

Click the button below to deploy the Logic App using the Default template. Your browser will be redirected to Microsoft Azure.

Deploy to Azure

(https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fwww%2Egstatic%2Ecom%2Fcloud%2Dmandiant%2Deng%2Dintegrations%2Fthreat%5Fintel%2Fazur

If you have used the Dependency template earlier to deploy the dependencies, you should specify the same values for **Key Vault Name**, **Key Vault Key Name**, **Storage Account Name**, and **Storage Container Name** for both the Dependencies template and the Default template.

Authorize Connections

Once deployment is complete, you must authorize each connection.

1. Microsoft Graph Security API Connection
    a. Click the **microsoftgraphsecurity** API connection.
    b. Click **Edit API Connection** from the **General** menu.
    c. Click **Authorize.**
    d. Sign in to Azure Key Vault.
    e. Click **Save**.
2. Azure Blob Storage Connection
    a. Open the **azureblobContainer** API connection.
    b. Click **Edit API Connection** from the **General** menu.
    c. Provide a valid Azure Storage Account Access Key.
    d. Access Key can be found in the **Storage Account Resource** under **Access Keys**.
    e. Click **Save**.

Assign Access Policy

Assign an access policy on the Key Vault for the playbook to fetch the secret key.

1. Select the **Key Vault Resource** where you have stored the secret key.
2. Click on **Access Policies**.
3. Click on **Create**.
4. Under the **Secret Permissions** column, select **Get & List** from **Secret Management Operations**.
5. Click **Next** to go to the **Principal** tab and choose your deployed playbook name.
6. Click **Next** and leave application tab open.
7. Click **Review and create**.
8. Click **Create**.

Configure the Logic App Settings

Important Considerations for Defender ATP
If you are using the Logic App to ingest indicators into Defender ATP, please note that Microsoft enforces a 15000 indicator per tenant limit. Consider using the filtering options provided in the Logic App to reduce the number of indicators ingested.

1. Complete the mandatory fields: **Resource group**, **API Key**, secret keys, i.e., **Key Vault Name**, **Key Vault Key Name**, and **Storage Account Name**.

    Do not configure a Key Vault or Storage Account prior to deployment. This will be provisioned automatically as part of the application deployment. If you would like to use an existing Storage Account, use the storage account name in the below configuration field. Ensure the container name is unique and not in use within that storage account.

2. (Optional) Change/configure the remaining fields to suit your needs. Each setting is described in the **table** below.
3. Click **Review + Create**. The Logic App is validated by Azure.
4. Once validation completes, click **Create**. The Logic App is deployed to your Azure instance.

An initial failure of the logic app is common. This is due to the fact that other resources are not yet available to the logic app (i.e., Key Vault is not fully deployed before the logic app attempts to run).

The following table lists the logic app settings API parameters, whether they are mandatory or optional, their default values, and descriptions.

Deployment Parameter	Mandatory or Optional	Default Value	Description
Key ID	Mandatory	None	The value of your Mandiant API Key
Secret Key	Mandatory	None	The value of your Mandiant API Key Secret
Azure Key Vault Key Name	Mandatory	None	The name of the Azure Key Vault to use for storing the API Secret Key
Indicator Expiration	Optional	30	Defines the number of days after ingestion that an indicator will be considered valid by Microsoft
Minimum Confidence Score	Optional	80	Defines the minimum Mandiant Indicator Confidence Score to ingest
Exclude Open-Source Intelligence	Optional	True	Defines if open-source indicators should be ingested
Lookback Interval	Optional	-30	Defines the number of days in the past to start ingesting indicators, based on an indicator's last_updated date value
Execution Interval	Optional	6	The frequency, in hours, that the Logic App will check Mandiant for new or updated indicators
Default Action	Optional	alert	The action that Microsoft products should trigger when the indicator is matched in the system, either alert, or block
Target Application	Optional	Azure Sentinel	The Microsoft application to make indicators visible in, either Azure Sentinel or Microsoft Defender ATP
Include Uncategorized	Optional	False	Defines if indicators without an attributed category are ingested or not
Hash Type	Optional	md5	The hash type to use for indicators of type file, either md5, sha1, or sha256
Include IPv4	Optional	True	Defines if IPv4 indicators should be ingested
Include FQDN	Optional	True	Defines if FQDN indicators should be ingested
Include URL	Optional	True	Defines if URL indicators should be ingested
Include File Hash	Optional	True	Defines if File Hash indicators should be ingested
Storage Account Name	Mandatory	None	The Name of an Azure Blob Storage Account to be created. Hosts the Storage Container
Storage Container Name	Mandatory	mandiant-threat-intel	The Container Name used to store the last runtime to support periodic updates