MICROSOFT AZURE SENTINEL INTEGRATION WITH SECURITY VALIDATION

This integration provides the following benefits:

- Validates that security tools are writing log events to Microsoft Azure Sentinel to ensure compliance with security policies and regulations
- Collects events generated by security tools that write to Microsoft Azure Sentinel to test the efficacy and configuration of security controls using Security Validation Kobs

Use this document to configure the integration using one of the following methods:

- MSI (Supported and recommended for new integration configurations)
- Legacy (Supported for existing integration configurations)

Configure MSI Integration

This method is the recommended approach for configuring new integrations in Security Validation.

If you're looking for raw event data, you should implement the **Microsoft Azure Log Analytics (https://docs.mandiant.com/home/msv-microsoft-azure-log-analytics)** integration. The integration described here, for Microsoft Azure Sentinel, addresses alerts associated with that event data.

Prerequisites

Information to gather before you start:

- Identify the Client ID unique to your application.
- Identify the Client Secret unique to your application.
- Identify the Tenant ID.
- Identify the Workspace ID.

Prepare Technology

- Identify or create credentials to access Sentinel with read access, at minimum.
- Add the Microsoft Sentinel Reader role to the app that you are creating and registering.
- Verify you have access to the Log Analytics API with Data.Read permission.
- Identify the following values in the Azure Web portal:

  These values are generated when you configure Azure Log Analytics.

  - Client ID
  - Client Secret
  - Tenant ID
  - Workspace ID

- Set up Tables in Log Analytics.

  Queries in the Azure Sentinel integration will error if corresponding Tables are not configured in Log Analytics. For example, the default Malicious DNS Action Query in the integration needs the DnsEvents table in Log Analytics to be configured.

Access the Client ID, Client Secret, Tenant ID, and Workspace ID

If you do not already know the values required to add the Azure Sentinel integration, you must locate them in the Azure portal.

- Refer to the **Microsoft documentation** (https://docs.microsoft.com/en-us/azure/active-directory-b2c/tutorial-register-applications?tabs=app-reg-ga) for further assistance identifying these values.
- If you have already noted the Client ID, Client Secret, Tenant ID, and Workspace ID for the **Microsoft Azure Log Analytics** (https://docs.mandiant.com/home/msv-microsoft-azure-log-analytics) Integration, you can reuse those values for the Azure Sentinel Integration.

1. In the Azure Log Analytics portal, take note of your **Workspace ID**.
2. In the Active Directory portal, take note of your **Tenant ID**.
3. In the Active Directory portal, navigate to **App registrations** > **New registration**.
4. Enter the required registration information.
    a. Take note of the **Client ID**.
    b. The required **Redirect URI** field can be set to your Director's URL.
5. Navigate to the **Certificates & Secrets** page.
6. Create a new client secret and take note of the value.

Add the Data.Read API Permission

1. In the Azure Log Analytics portal, navigate to the **API Permissions** page.
2. Add Log Analytics **Data.Read** permission.
3. Get administrator approval for the application.

API Calls

The following API calls are used by the Validation Platform.

Purpose	Call
Auth	https://login.microsoftonline.com/{tenent_id}/oauth2/token For Azure Government (GovCloud): https://login.microsoftonline.us/{tenent_id}/oauth2/token
Query Log Analytics	https://api.loganalytics.io/v1/workspaces/{workspace_id}/query For Azure Government (GovCloud): https://api.loganalytics.us/v1/workspaces/{workspace_id}/query

Configure Security Validation

1. Go to **Settings** > **Integrations**.
2. From the Preview Direct Integrations table, click **Add Integration** > **Microsoft Azure Sentinel**.

    You can add this as either a Direct or Remote Integration.

3. Enter a meaningful **Integration Name**.
4. Optional: From the **Proxy** drop-down, choose a proxy profile if one is available. If one isn't available and all

outbound connections go through a proxy, first, set up a **Proxy Rule** (https://docs.mandiant.com/home/msv-proxy-rules).

5. For the **Host**, enter the appropriate value depending on your Azure Sentinel environment:
   - api.loganalytics.io for standard Azure environments
   - api.loganalytics.us or Azure Government (GovCloud) environments
6. Enter a **Port** value. The default is **443**.
7. Enter **Client Id** and **Client Secret**.
8. Enter **Tenant Id** and **Workspace Id**.
9. Optional: Check **Verify Ssl** if you want this verification done for requests to an upstream server.
10. Optional: Change the **Timeout** value if you want a different frequency of requests to an upstream server. The default is **30** (seconds).
11. Optional: Add or modify any values in the **Query** section, as needed. The default queries are given in the web interface.
12. Optional: Modify the **Field Map** values, as necessary.

    - Each field map box can hold a JSON-formatted comma-separated list of columns returned by the API to be considered for each field when translating into the normalized event object format. Example: description could be configured to be 'msg_s' or 'SyslogMessage' in some environments. The field map tries both if set to: ['msg_s','SyslogMessage'] and whichever matches first is the column that is used.
    - When configuring an integration in Security Validation, you can assign additional host values in the Field Map settings. If none of the assigned fields return a valid host name, Network Actions may miss matched events from the third-party technology. Additional hosts values helps ensure the likelihood of a match between the two environments.

13. Optional: Expand **Advanced options** and update the information as necessary.
    a. Update **Query Time** and **Delay Time**.

    The **Query time** is the amount of time (minutes) before and after the query runs that the platform looks for events, while the **Delay time** is the amount of time (minutes) that the platform waits to run the first query after a Job Action starts. For example, you configure your integration with the following values: **Query time** = 5, **Query interval** = 30 seconds, and **Delay time** = 0. When a Job Actions starts at 12:00:00, the first time the query runs, the platform looks for events from 11:55:00 to 12:00:00. Then 30 seconds later, it looks for events from 11:55:30 to 12:00:30. This interval continues, with the last query looking from 12:00:00 to 12:05:00. If you instead configured the **Delay time** to equal 10, it would run the same query, but it wouldn't start that query until 12:10:00.

    If your monitors are set to run more frequently than the query time, this configuration impacts the pass/fail results for AEDA monitors.

    b. Update **Query Interval** (seconds).
    c. Configure correlation queries:
       i. Select **Correlation Query Enabled** and fill in the **Correlation Query**.
       ii. Modify the **Correlation Query Interval**, if necessary (minutes).
    d. Select **Discover network devices automatically**, the default and recommended option.

    If unselected, reported events won't include product information for any matching network security technology.

e. Select **Save Suspicious Events**.
f. Modify the **Event Time Adjustment** (seconds). The default is **0**.
g. Modify the **Limit** value if you need to prevent a flood of results. This value is set to **10000** by default. This limit applies to both events and alerts individually, so if you set it to **10**, you can still see a maximum of 10 events and 10 alerts.

14. Click **Save**.

Verify connectivity

1. Go to **Settings > Integrations**.
2. From the Direct Integrations table, click ⋮ **> Test** to verify that:
   - The Director can communicate with the integration host on the port and protocol specified.
   - The integration credentials are valid and working.

For more information on setting up queries, see **Manage Integrations** (https://docs.mandiant.com/home/msv-managing-integrations).

Configure Legacy Integration

This document applies to Classic/Legacy Integrations. You may continue to use these integration configurations. While no active development is happening for these integrations, we continue to provide Classic/Legacy Integrations in the product. You do not have to move to MSI Integrations. If your support engineer or TSC recommends or you choose to move to MSI Integrations, you can take advantage of the latest features and functionality. For more information, see the MSI Integration documentation in the **Integrations Overview** (https://docs.mandiant.com/home/msv-integrations-overview).

If you're looking for raw event data, you should implement the **Microsoft Azure Log Analytics** (https://docs.mandiant.com/home/msv-microsoft-azure-log-analytics) integration. The integration described here, for Microsoft Azure Sentinel, addresses alerts associated with that event data.

Prerequisites

Information to gather before you start:

- Identify the Client ID unique to your application.
- Identify the Client Secret unique to your application.
- Identify the Tenant ID.
- Identify the Workspace ID.

Configure the Azure Sentinel Integration

1. Go to **Settings** > **Integrations**.
2. Click **Add Integration** > **Azure Sentinel**.

   You can add this as either a Local or Remote Integration.

3. From the **Host** drop-down list, select the appropriate value depending on your Azure Sentinel environment:
   - The entry ending in **.io** for standard Azure environments
   - The entry ending in **.us** for Azure Government (GovCloud) environments

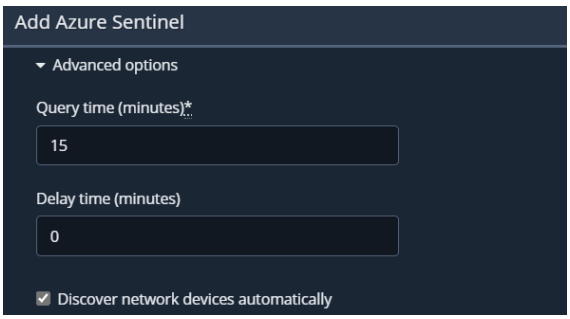
4. Enter **Client ID** and **Client Secret**.

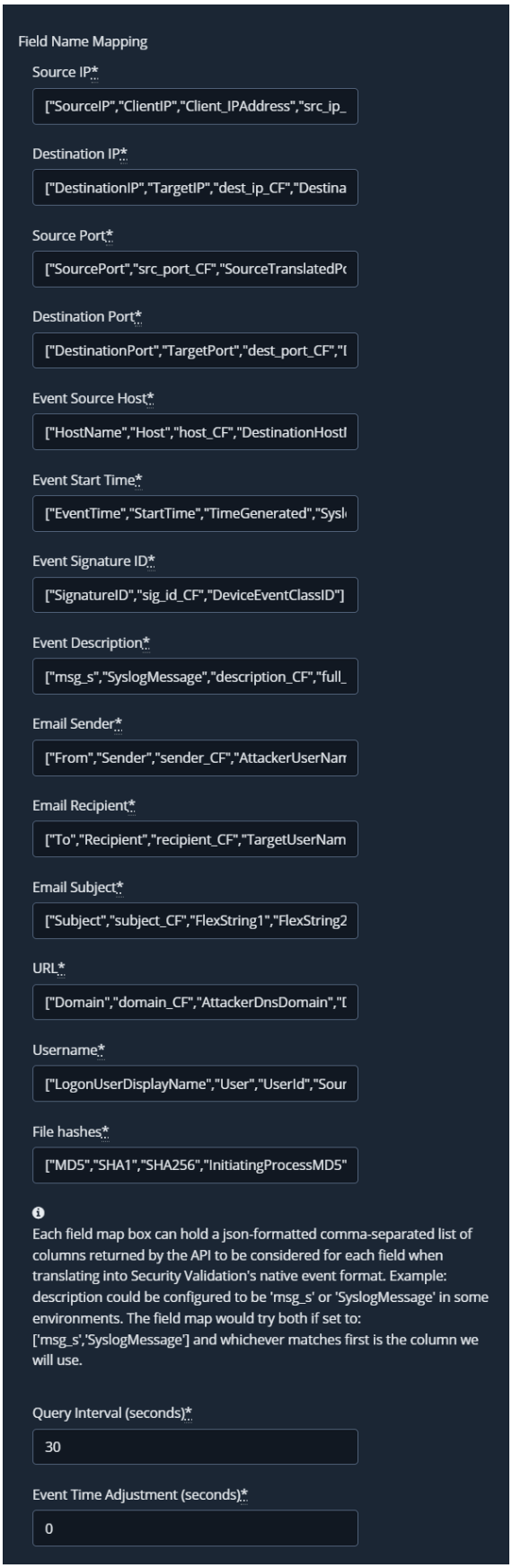
5. Enter **Tenant ID** and **Workspace ID**.
6. Expand **Advanced options** and update the information as necessary.
    a. (Optional) Update **Query time (minutes)** and **Delay time (minutes)**.

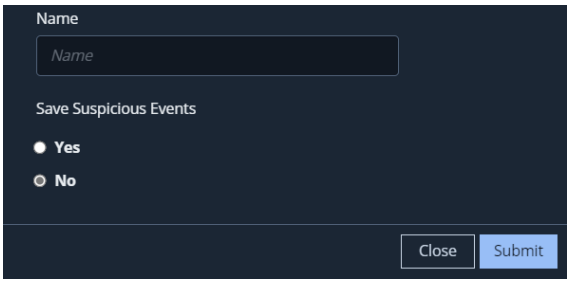
> The **Query time** is the amount of time (minutes) before and after the query runs that the platform looks for events, while the **Delay time** is the amount of time (minutes) that the platform waits to run the first query after a Job Action starts. For example, you configure your integration with the following values: **Query time** = 5, **Query interval** = 30 seconds, and **Delay time** = 0. When a Job Actions starts at 12:00:00, the first time the query runs, the platform looks for events from 11:55:00 to 12:00:00. Then 30 seconds later, it looks for events from 11:55:30 to 12:00:30. This interval continues, with the last query looking from 12:00:00 to 12:05:00. If you instead configured the **Delay time** to equal 10, it would run the same query, but it wouldn't start that query until 12:10:00.

> If your monitors are set to run more frequently than the query time, this configuration impacts the pass/fail results for AEDA monitors.

    b. (Optional) Select **Discover network devices automatically**.
    c. Modify **Field Name Mapping**for the following, as necessary:
        - **Source IP**
        - **Destination IP**
        - **Source Port**
        - **Destination Port**
        - **Event Source Host**
        - **Event Start Time**
        - **Event Signature ID**
        - **Event Description**
        - **Email Sender**
        - **Email Recipient**
        - **Email Subject**
        - **URL**
        - **Username**
        - **File hashes**
    d. Modify the **Query Interval (seconds)** and **Event Time Adjustment (seconds)**, if necessary.
    e. (Optional) Assign a **Name**.
    f. (Optional) Choose **Yes** to **Save Suspicious Events**.
7. Click **Submit**.







Microsoft Azure Sentinel Integration - Advanced Options

Set up Proxy Assignment

If all outbound connections go through a proxy, you may want to set up a proxy definition and assignment for your integration. For information on setting up your proxy rules, see **Proxy Rules (https://docs.mandiant.com/home/msv-proxy-rules)**.

Verify connectivity

Click **Test** to verify that:

- The Director can communicate with the integration host on the port and protocol specified.
- The integration credentials are valid and working.