NetWitness® Platform

Microsoft Azure Graph Event Source Log Configuration Guide

NETWITNESS
Platform

Microsoft Azure Graph

Event Source Product Information:

Vendor: Microsoft

Event Source: Azure Graph API

Versions: API v1.0

NetWitness Product Information:

Supported On: NetWitness Platform 12.2 and later

> **Note:** Azure Graph API is supported from NetWitness Platform 11.5. However, NetWitness recommends you to update NetWitness Platform to the latest version.

Event Source Log Parser: azure

Collection Method: Plugin Framework

Event Source Class.Subclass: Host.Cloud

Contact Information

NetWitness Community at https://community.netwitness.com contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

License Agreement

Third-Party Licenses

Note on Encryption Technologies

Distribution

Miscellaneous

November 2024

Contents

About Microsoft Graph API

Microsoft Graph is the gateway to data and intelligence in Microsoft 365. It provides a unified programmability model that you can use to access the tremendous amount of data in Microsoft 365, Windows 10, and Enterprise Mobility + Security. Use the wealth of data in Microsoft Graph to build apps for organizations and consumers that interact with millions of users. The power of Microsoft Graph lies in the easy navigation of entities and relationships across different services exposed on a single Microsoft Graph REST endpoint. For more information, please refer https://docs.microsoft.com/en-us/graph/api/overview?view=graph-rest-1.0.

NetWitness Platform captures logs and security alerts from Microsoft Azure through the Microsoft Graph API and parses the collected information into metas.

IMPORTANT: Links to Microsoft website provided in this document are subject to change by Microsoft.

NetWitness Supported Event Sources and Permissions Information

The below table provides information about Microsoft Azure event sources and their permission details.

Microsoft Azure Event Sources	Permission Details	URL (Required during NetWitness Plugin Configuration)
Directory Audit	https://docs.microsoft.com/en-us/graph/api/directoryaudit-list?view=graph-rest-1.0&tabs=http	https://graph.microsoft.com/v1.0/auditLogs/directoryAudits?$filter=activityDateTime ge {starttime} and activityDateTime lt {endtime} **Note:** To ensure a successful plugin configuration, it is recommended that you copy and paste the same URL provided. This will ensure accuracy and prevent potential errors.
Sign-Ins	https://docs.microsoft.com/en-us/graph/api/signin-list?view=graph-rest-1.0&tabs=http	https://graph.microsoft.com/v1.0/auditLogs/signIns?$filter=createdDateTime ge {starttime} and createdDateTime lt {endtime} **Note:** To ensure a successful plugin configuration, it is recommended that you copy and paste the same URL provided. This will ensure accuracy and prevent potential errors.

Microsoft Azure Event Sources	Permission Details	URL (Required during NetWitness Plugin Configuration)
Risk Detections	https://docs.microsoft.com/en-us/graph/api/riskdetection-list?view=graph-rest-1.0	https://graph.microsoft.com/v1.0/identityProtection/riskDetections?$filter=activityDateTime ge {starttime} and activityDateTime lt {endtime} **Note:** To ensure a successful plugin configuration, it is recommended that you copy and paste the same URL provided. This will ensure accuracy and prevent potential errors.
Security Alerts	https://docs.microsoft.com/en-us/graph/api/alert-list?view=graph-rest-1.0&tabs=http	`https://graph.microsoft.com/v1.0/security/alerts?$filter=lastModifiedDateTime ge {starttime} and lastModifiedDateTime lt {endtime}` **OR** If you want to use latest v2 API url, the value is `https://graph.microsoft.com/v1.0/security/alerts_v2?$filter=createdDateTime ge {starttime} and createdDateTime lt {endtime}` **Note:** To ensure a successful plugin configuration, it is recommended that you copy and paste the same URL provided. This will ensure accuracy and prevent potential errors.
Azure Sentinel Incidents	https://learn.microsoft.com/en-us/azure/sentinel/roles#role-and-permissions-recommendations. Ex:Microsoft Sentinel Reader	https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.OperationalInsights/workspaces/{workspaceName}/providers/Microsoft.SecurityInsights/incidents?api-version=2023-02-01&$filter=properties/lastModifiedTimeUtc ge {starttime} and properties/lastModifiedTimeUtc le {endtime} &$orderby=properties/lastModifiedTimeUtc **Note:** To ensure a successful plugin configuration, it is recommended that you copy and paste the same URL provided. This will ensure accuracy and prevent potential errors.
Microsoft Defender XDR incidents	List incidents API in Microsoft Defender XDR - Microsoft Defender XDR | Microsoft Learn	https://api.security.microsoft.com/api/incidents?$filter=lastUpdateTime ge {starttime} and lastUpdateTime lt {endtime} **Note:** To ensure a successful plugin configuration, it is recommended that you copy and paste the same URL provided.This will ensure accuracy and prevent potential errors.

To configure Microsoft Azure Graph, you must complete the following tasks

I. Configure the Microsoft Azure Graph Event Source

II. Set Up the Microsoft Azure Graph Plugin Event Source in NetWitness Platform

Configure the Microsoft Azure Graph Event Source

This section describes how to use the Azure Management Portal to register an application in Microsoft Entra ID (formerly known as Azure Active Directory), and create a key.

Steps to register an application in Microsoft Entra ID:

1. Go to Register a New Application Using the Azure Portal and follow the instructions to register an application.
2. Locate the API Permissions section for your registered application, and under the API permissions, click **Add a permission**.
3. Assign the Application Permissions to the registered application.

   Note: Set the application permissions the same as the required permissions given under permission type **Application**. To know more, see **Permission Details** in NetWitness Supported Event Sources and Permissions Information.

4. Click **Grant admin consent for {directory}**.

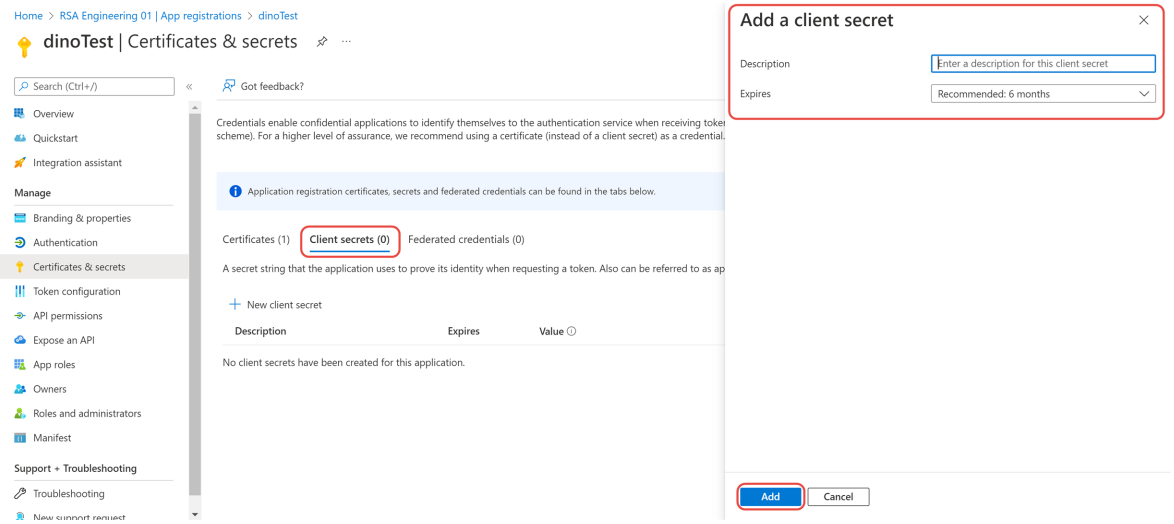
   Note: Only Azure Admin can grant consent for {directory}. If you are not an admin, consult the Azure Admin.

   Note: Any other event source which can send logs to Azure Graph API can be collected. Make sure to set the required permissions in the Azure application before collecting. Then, configure the plugin in NetWitness. Create a custom parser or contact NetWitness support to parse the collected logs.

5. Create a secret key or certificate to authenticate Microsoft Azure application.
   - Create a Client secret:

     In the left menu bar, click **Certificates & secrets**, then click **New client secret**. Add new client secret information and click **Add**.

     IMPORTANT: Azure only displays the client secret at the time you generate it. You cannot navigate back to this page and retrieve the client secret later. Make sure to copy and save this key, as it is needed for further configuration.



- Create a Certificate:

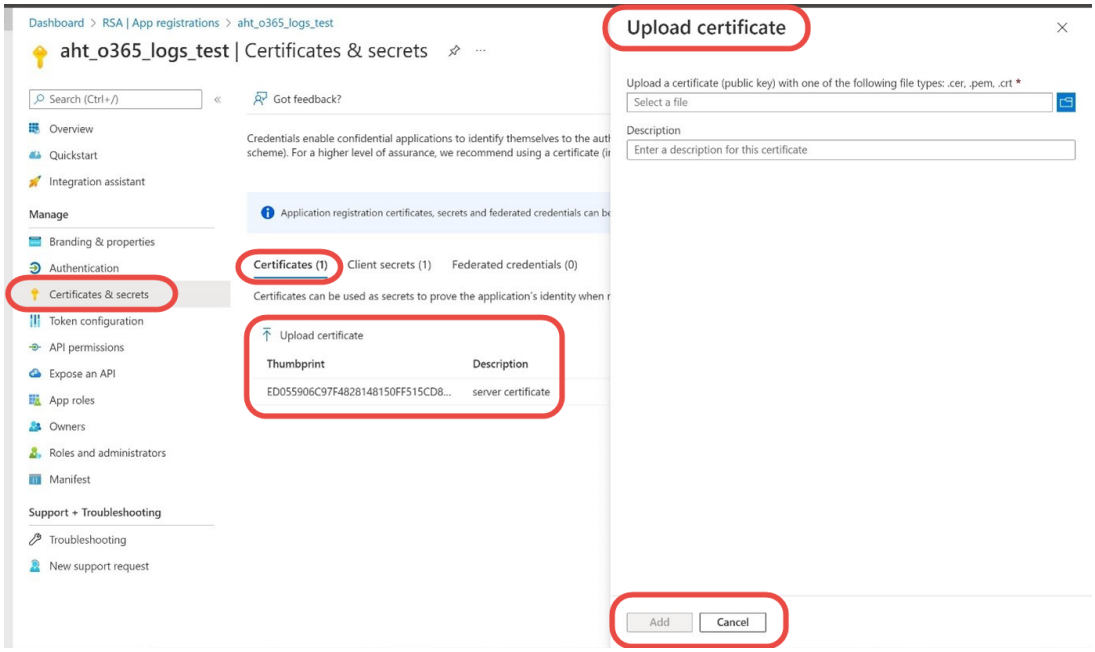
  1. Create a certificate and private key for authentication. Follow the steps in the link below. https://github.com/AzureAD/azure-activedirectory-library-for-python/wiki/Client-credentials#client-credentials-with-certificate.

     IMPORTANT: When you execute Create a certificate request command, make sure that you pass a blank value for `A challenge password []:`

     IMPORTANT: Keep the `server.pem` file securely as it is required to provide as an input to configure NetWitness plugin.

  2. In **Microsoft Entra ID (formerly known as Azure Active Directory)** app, go to **Certificates & secrets** > **Certificates** > **Upload certificate**.
  3. On the **Upload certificate** dialog,
     a. Select the `server.crt` certificate that you have created in step 1.
     b. Provide a short description in the **Description box** and click **Add**. A Thumbprint is created

after the certificate is uploaded successfully.



IMPORTANT: `thumbprint` and the private key inside `server.pem` file (created at step 1) should be given as inputs to configure Netwitness Plugin.

Set Up the Microsoft Azure Graph Plugin Event Source in NetWitness Platform

In NetWitness Platform, perform the following tasks

I. Deploy Microsoft Azure Graph Files from NetWitness Platform.

II. Configure the Microsoft Azure Graph Event Source in NetWitness Platform.

Deploy Microsoft Azure Graph Files from NetWitness Platform

Microsoft Azure Graph plugin requires resources available in Live to collect logs.

To deploy the required content from Live:

1. In the NetWitness Platform menu, select (Configure).
   The **Live Content** tab is displayed.
2. Browse the Live collection file as **msazuregraph**, using **Log Collector** as the Resource Type.
3. Select **msazuregraph** from the list and click **Deploy** to deploy it to appropriate Log Collectors, using the Deployment Wizard.
4. Deploy Azure Parser which is the dependent parser for this plugin. Make sure you deploy it to appropriate Log Decoders.

   Note: If you are using a remote Virtual Log Collector for collection, then deploy the plugin log collection to the remote Virtual Log Collector as well as the Log Decoder.

For more details, see the Add or Update Supported Event Source Log Parsers topic.

Configure the Microsoft Azure Graph Event Source in NetWitness Platform

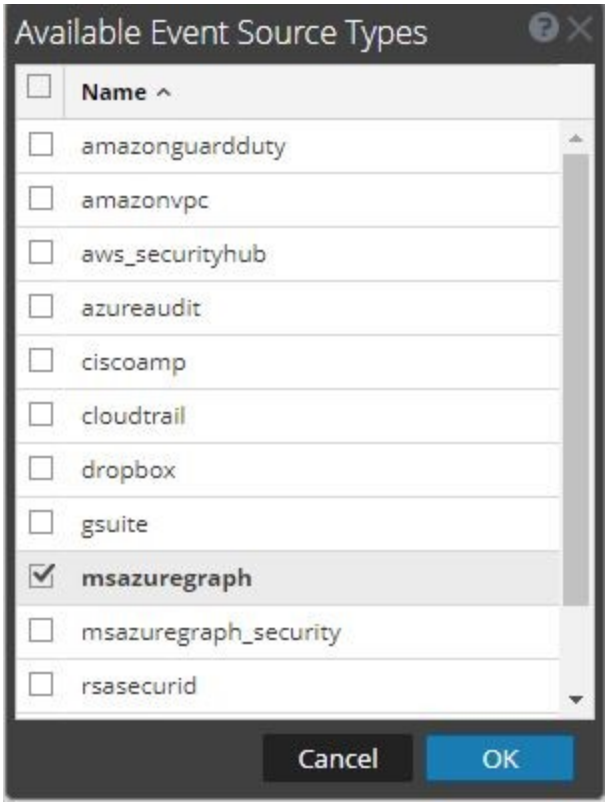
To configure the Microsoft Azure Graph Event Source:

1. In the NetWitness Platform menu, select (Admin) > **Services**.
2. In the **Services** grid, select a Log Collector service, and from the **Actions** menu, choose **View** > **Config**.
3. In the **Event Sources** tab, select **Plugins/Config** from the drop-down menu.

The **Event Categories** panel displays the File event sources that are configured, if any.

4. In the **Event Categories** panel toolbar, click +.

   The **Available Event Source Types** dialog is displayed.

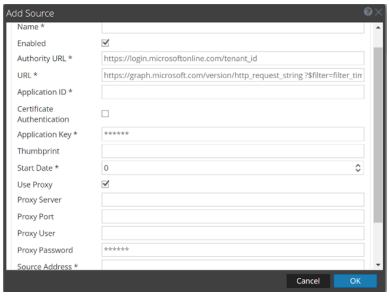


5. Select **msazuregraph** from the list, and click **OK**.

   The newly added event source type is displayed in the **Event Categories** panel.

6. Select the **new type** in the **Event Categories** panel and click + in the **Sources** panel toolbar.

   The **Add Source** dialog is displayed.



7. Define parameter values, as described in Microsoft Azure Graph Collection Configuration Parameters.
8. Click **Test Connection**.

The result of the test is displayed in the dialog box. If the test is unsuccessful, edit the device or service information and retry.

Note: The Log Collector takes approximately 60 seconds to return the test results. If it exceeds the time limit, the test times out, and NetWitness Platform displays an error message.

9. If the test is successful, click **OK**.

   The new event source is displayed in the **Sources** panel.

10. Repeat steps 4–9 to add another Microsoft Azure Graph plugin instance.

Microsoft Azure Graph Collection Configuration Parameters

The following table describes the configuration parameters for the Microsoft Azure Graph integration with NetWitness Platform.

Note: Items that are followed by an asterisk (*) are required.

Basic Parameters

Name	Description
Name *	Enter an alpha-numeric, descriptive name for the source. This value is only used for displaying the name on this screen.
Enabled	The box is selected by default. Select the box to enable the event source configuration to start collection.
Authority URL *	Enter `https://login.microsoftonline.com/<tenant-id>`. Replace tenant-id with your tenant ID.
URL *	Request URL collects events. See NetWitness Supported Event Sources and Permissions Information to obtain the correct URLs. Expected URL value format is: `https://graph.microsoft.com/v1.0/<http request string>?$filter=filter_timestamp ge {starttime} and filter_timestamp lt {endtime}` *filter_timestamp* is the value used for filtering events. It must be one of timestamp fields in the expected log. *{starttime}* and *{endtime}* will be replaced with values based on startdate and current time during plugin collection. **IMPORTANT:** To ensure a successful plugin configuration, it is recommended that you copy and paste the provided URL. This will ensure accuracy and prevent potential errors.
Application ID *	The Client ID is found in the Azure Application **Configure** tab. Scroll down until you see it.
Certificate Authentication	Select the box to enable the use of Certificate Authentication to collect the events. Do not enable Certificate Authentication if you opt for Client Secret Authentication.

Name	Description
Application Key *	Enter the Client Secret or Private key, • If you opt for Certificate Authentication, enter full content from the file `server.pem` For more information, see Create a Certificate. • If you opt for Client Secret Authentication, enter the client secret. For more information, see Create a Client Secret.
Thumbprint	Provide thumbprint from the ".cert" file as input if you opt for Certificate Authentication otherwise leave empty. For more information, see Create a Certificate.
Start Date*	Choose the date from which to start collecting. This parameter defaults to the current date. Enter a value from 0 to 7, indicating how many days in the past from which to start collection. **Note:** Do not edit the Start Date value of a running graph plugin instance. This value is used for bookmarking purpose to avoid collection of duplicate logs. To start from a different start date, create a new plugin event source.
Use Proxy	Check to enable proxy.
Proxy Server	If you are using a proxy, enter the proxy server address.
Proxy Port	Enter the proxy port.
Proxy User	Username for the proxy (leave empty if using anonymous proxy).
Proxy Password	Password for the proxy (leave empty if using anonymous proxy).
Source Address *	IP address that is provided to the Azure Graph plugin instance. Logs from this event source will be collected using this device IP. **Note:** This is an arbitrary IP address chosen by the user. This value has no bearing on the collection of logs: its value is captured by the device.ip meta key, and can help you to query or group events collected by a particular instance of the plugin.
Test Connection	Checks the configuration parameters specified in this dialog to make sure they are correct.

Note: Please avoid using special characters in the **Proxy User** and **Proxy Password** sections.

Advanced Parameters

Parameter	Description
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is **180**. For example, if you specify 180, the collector schedules a polling of the event source every 180 seconds. If the previous polling cycle is still underway, it will wait for it to finish that cycle. If you have a large number of event sources that you are polling, it may take longer than 180 seconds for the polling to start because the threads are busy.
Max Duration Poll	Maximum duration, in seconds, of a polling cycle. A zero value indicates no limit. The default is set to **600**.
Max Events Poll	The maximum number of events per polling cycle (how many events collected per polling cycle).
Max Idle Time Poll	Maximum duration, in seconds, of a polling cycle. A zero value indicates no limit.
Command Args	Optional arguments to be added to the script invocation.
Debug	**Caution:** Enable debugging (set this parameter to **On** or **Verbose**) only if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. After changing this value, the change takes effect immediately (no restart required). The debug logging is **Verbose**, so limit the number of event sources to minimize performance impact.
SSL Enabled	The check box is selected by default. Uncheck this box to disable SSL certificate verification.
Trail By (in minutes)	Specifies the lag between current time and log collection in NetWitness Plugin. The default value is 10 minutes. Range is set between 5 to 1440 minutes. You need to tune this value if you are seeing logs which are near to real time is missing in NetWitness Platform Investigation page. **Note:** It was observed that audit and sign-in logs may take sometime to appear in Azure portal, hence adjusting the trail-by value is important to not miss any logs.

Parameter	Description
Scopes	Scopes used to generate token for API access using Microsoft MSAL. Default value is `https://graph.microsoft.com/.default`. **Note:** Scope for sentinel_incidents is https://management.azure.com/.default Please refer https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-on-behalf-of-flow for more details. **Note:** Scope for Microsoft Defender XDR incidents collection is https://api.security.microsoft.com/.default **Note:** If there are multiple values, include them as comma separated values.

Getting Help with NetWitness Platform

Self-Help Resources

There are several options that provide you with help as you need it for installing and using NetWitness:

- See the documentation for all aspects of NetWitness here: https://community.netwitness.com/t5/netwitness-platform/ct-p/netwitness-documentation.
- Use the **Search** and **Create a Post** fields in NetWitness Community portal to find specific information here: https://community.netwitness.com/t5/netwitness-discussions/bd-p/netwitness-discussions.
- See the NetWitness Knowledge Base: https://community.netwitness.com/t5/netwitness-knowledge-base/tkb-p/netwitness-knowledge-base.
- See the documentation for Logstash JDBC input plugin here: https://www.elastic.co/guide/en/logstash/current/plugins-inputs-jdbc.html.
- See Troubleshooting section in the guides.
- See also NetWitness® Platform Blog Posts.
- If you need further assistance, Contact NetWitness Support.

Contact NetWitness Support

When you contact NetWitness Support, please provide the following information:

- The version number of the NetWitness Platform or application you are using.
- Logs information, even source version, and collection method.
- If you have problem with an event source, enable **Debug** parameter (set this parameter to **On** or **Verbose**) and collect the debug logs to share with the NetWitness Support team.

Use the following contact information if you have any questions or need assistance.

NetWitness Community Portal	https://community.netwitness.com In the main menu, click **Support > Case Portal > View My Cases**.
International Contacts (How to Contact NetWitness Support)	https://community.netwitness.com/t5/support/ct-p/support
Community	https://community.netwitness.com/t5/netwitness-discussions/bd-p/netwitness-discussions

Feedback on Product Documentation

You can send an email to feedbacknwdocs@netwitness.com to provide feedback on NetWitness Platform documentation.