ReversingLabs Scanner for Microsoft Defender XDR

Version 1.0.0

Table of Contents

Document Version History

Version	Description	Date
1.0	Initial document creation for integration	2024-06-21

1. About the Scanner for Microsoft Defender XDR

The ReversingLabs Scanner for Microsoft Defender XDR is a utility for security teams to take advantage of the powerful file analysis capabilities of the Spectra suite of products. This solution is offered as a cloud-native offering, capable of being deployed within a customer's Azure tenant.

The ReversingLabs Scanner for Microsoft Defender integrates with Microsoft Defender APIs to enrich security incidents and scan files detected by Microsoft Defender for Endpoint. This powerful integration enables security teams to validate files identified by Microsoft Defender for Endpoint.

1.1 Features

- Reads events generated by the Microsoft Defender streaming API
- Identifies file hash entities in Microsoft Defender alerts and incidents
- Adds useful context and threat information relating to file hash entities, including:
    - A threat classification
    - The threat name
    - Anti-virus scanner detection rates

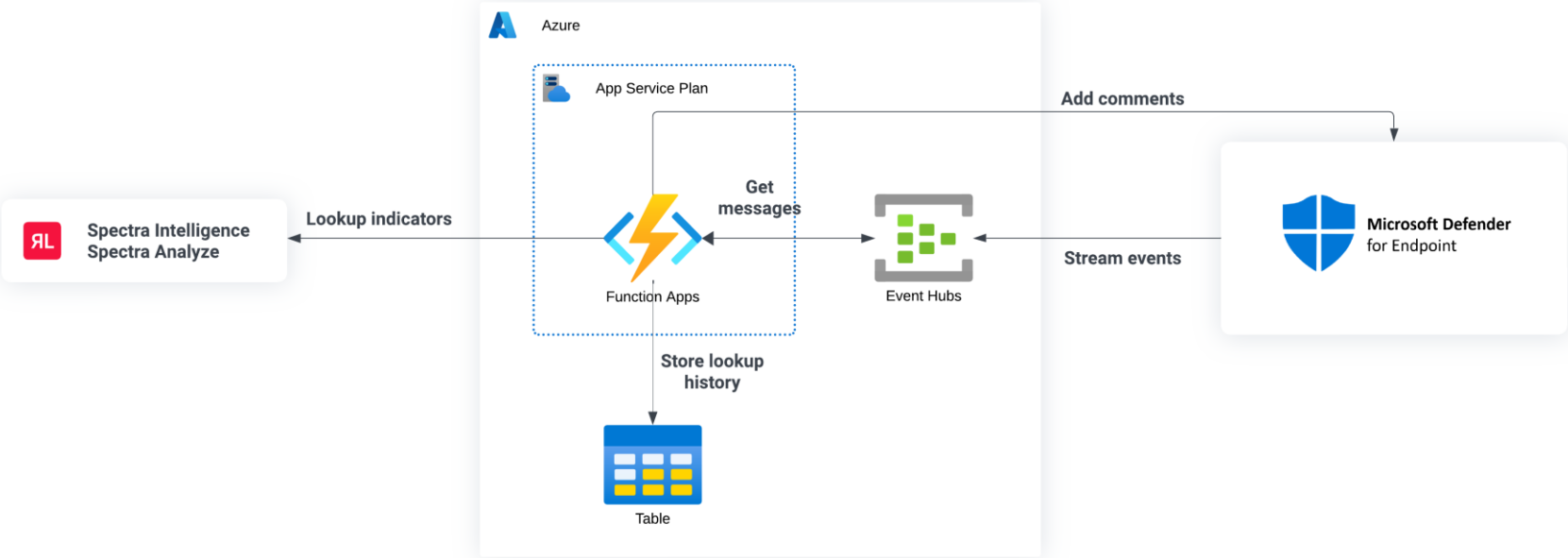
1.2 Pre-requisites

To use this solution, you must first:

- Have valid ReversingLabs Spectra Intelligence or Spectra Analyze API credentials
- Have an active Azure subscription
- Have active Microsoft Defender for Endpoint licenses

1.3 Architecture

The solution has been developed to use cloud-native resources in Microsoft Azure that can be individually deployed into a customer's tenant. The diagram on the next page visually illustrates a standard implementation of the solution:



1.4 Azure Resources

The following table describes the Azure resources deployed as part of this solution:

Resource Type	Description
Function App	The Function App scans Microsoft Defender for file hash entities to automatically enrich.
Event Hub	The Event Hub stores messages from the Microsoft Defender streaming API.
Storage account	The Storage Account contains the underlying Function App files and a Table for storing a lookup history.
Key Vault	Securely stores secrets and other configuration information.

1.5 API Permissions

This solution requires several API permissions to be able to read incidents and alerts generated in Microsoft Defender.

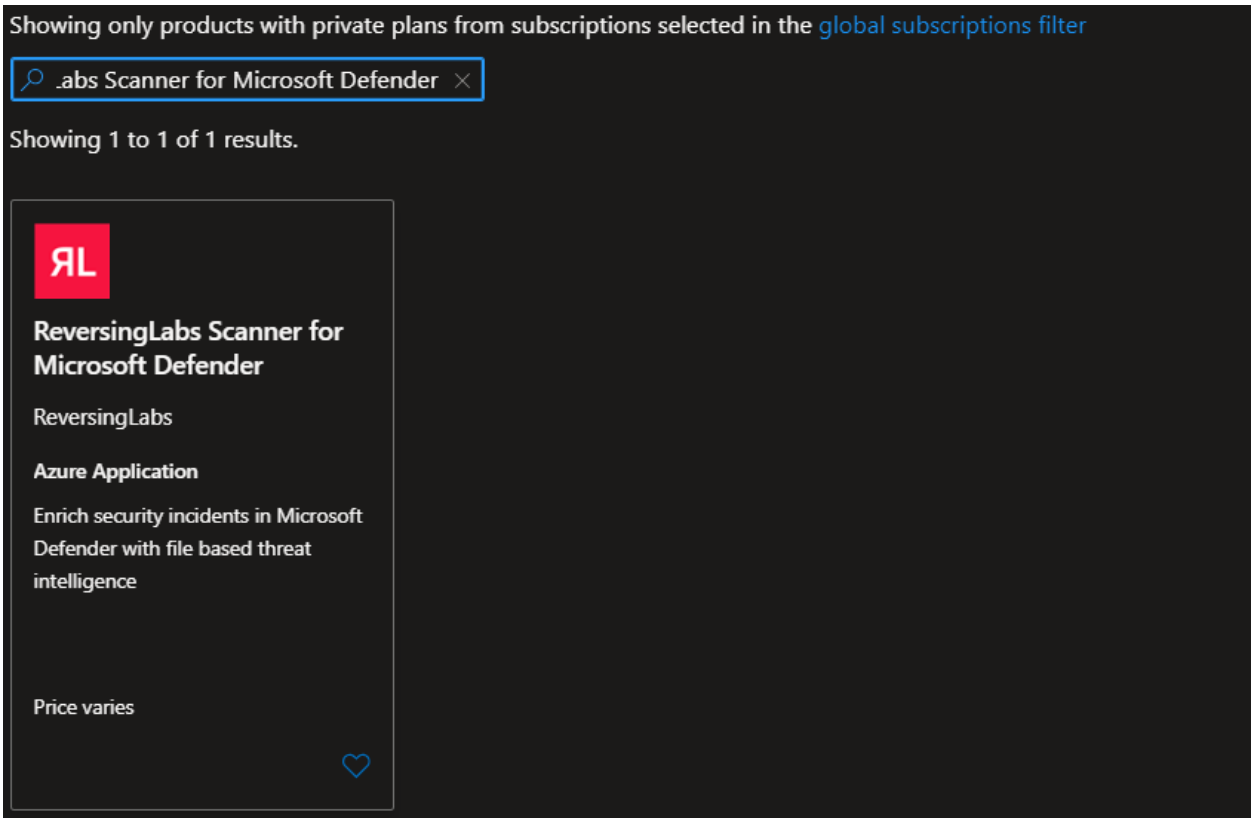
Scanner
A managed identity is created for the scanner function app during deployment. The following Microsoft Graph API permissions are required:

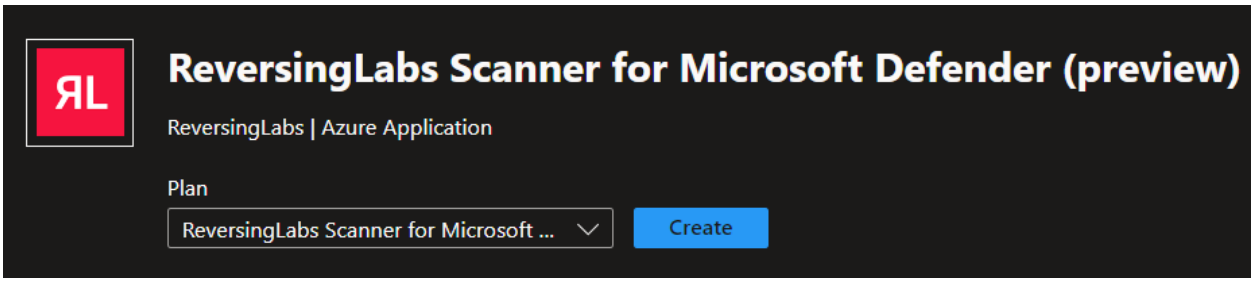
Permission	Type	Reason
Alert.ReadWrite.All	WindowsDefenderATP	Allows the application to read and write to Alerts
SecurityAlert.Read.All	Microsoft Graph	Allows the application to read and write to Alerts
Incident.ReadWrite.All	Microsoft Threat Protection	Allows the application to read and write to Incidents
SecurityIncident.ReadWrite.All	Microsoft Graph	Allows the application to read and write to Incidents

2. Installation

2.1 Marketplace Offer

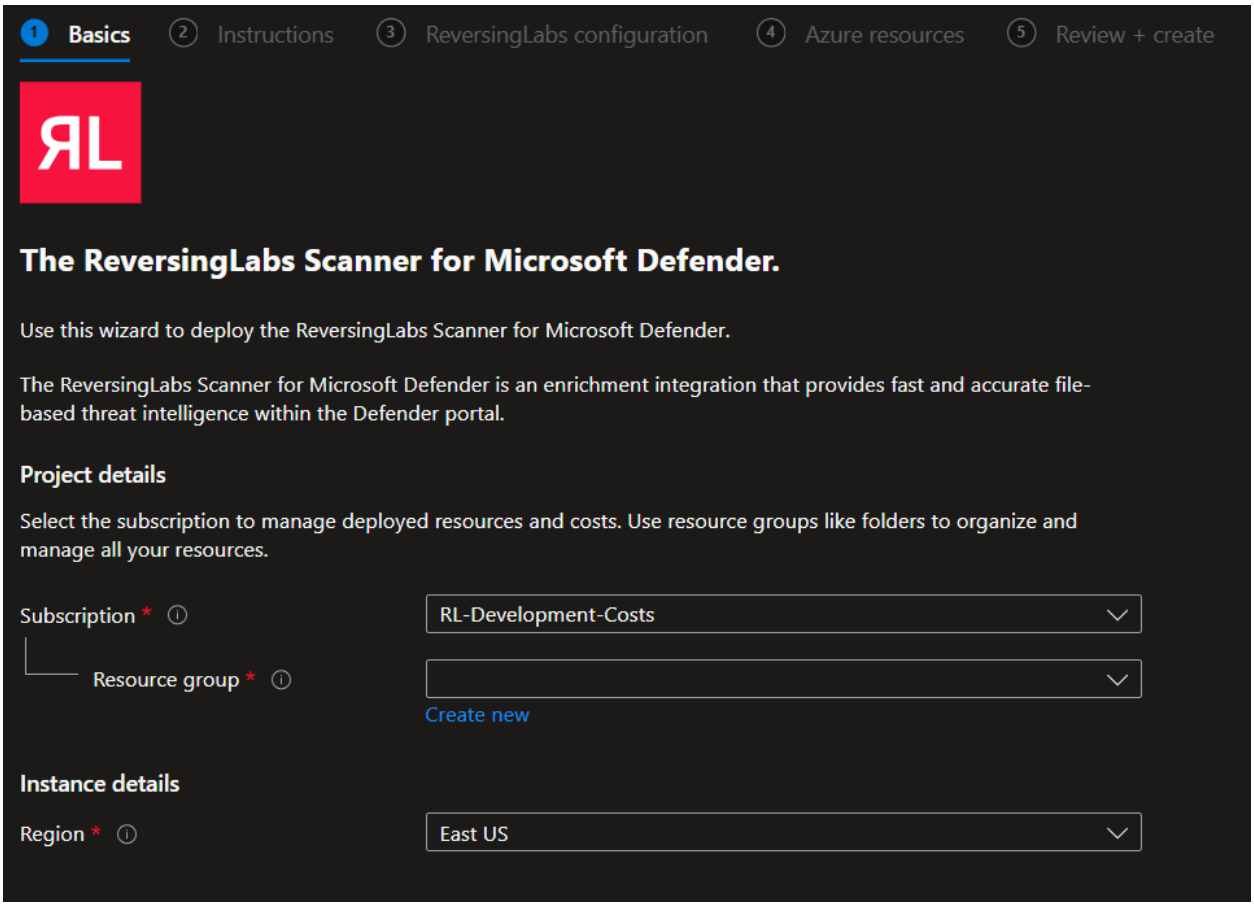
To install the ReversingLabs Scanner for Microsoft Defender, navigate to the Azure Marketplace and search for “ReversingLabs Scanner for Microsoft Defender”. Select the offering, then click “Create” next to the available plan:





2.2 Template Wizard

The template wizard will walk you through the installation process. First select the subscription, resource group, and region where the resources will be deployed:



The instructions step provides a link to this instruction guide.

The third step “ReversingLabs configuration” requires a selection of the ReversingLabs product that you would like to integrate with the scanner. Select one of or both options, then fill out the associated fields.

Basics | Instructions | 3 ReversingLabs configuration | 4 Azure resources | 5 Review + submit

Select the ReversingLabs product you wish to integrate with Defender.

NOTE: A valid ReversingLabs license is required to use this integration. See our website for details on how to acquire a license.

Spectra Analyze (formerly A1000) ☑

Spectra Intelligence (formerly TitaniumCloud) ☑

Spectra Intelligence Configuration

Spectra Intelligence Username *

Spectra Intelligence Password *

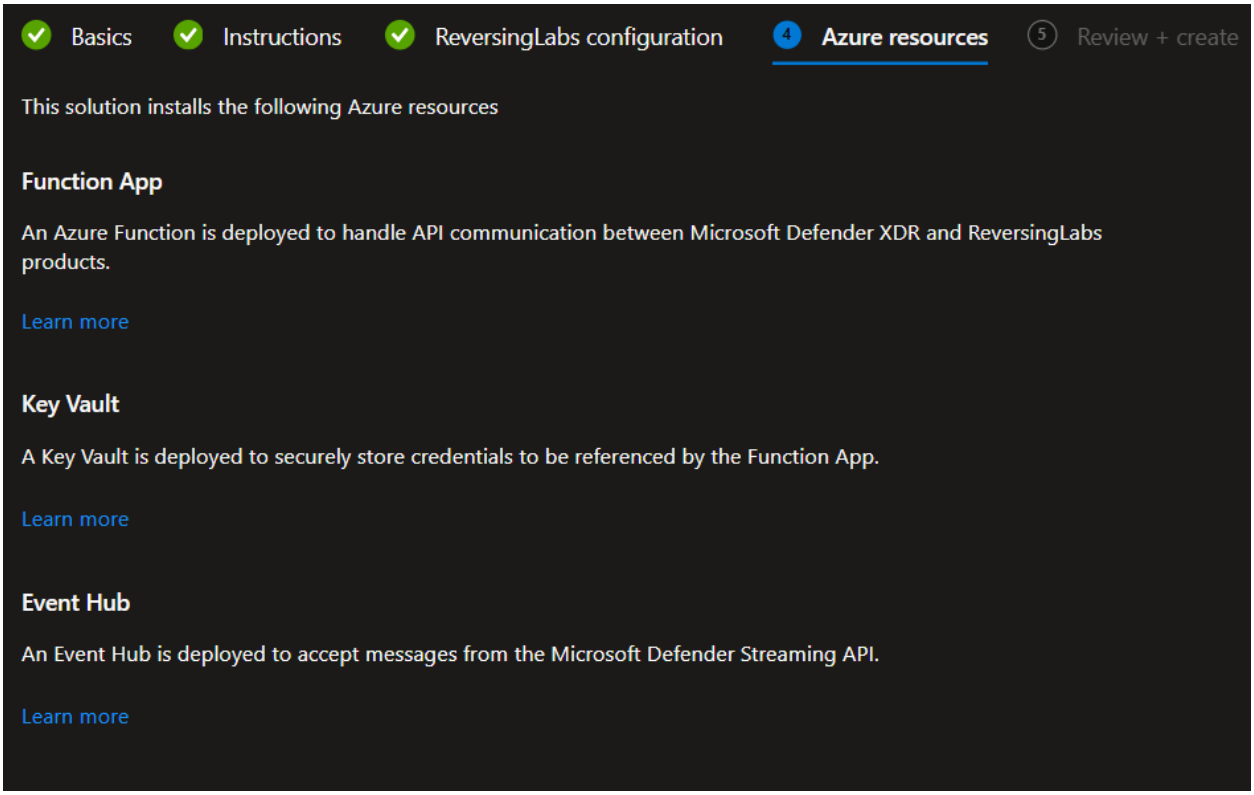
Confirm password *

Spectra Analyze Configuration

Spectra Analyze URL *

Spectra Analyze API Token *

The fourth step describes the Azure resources that are deployed as part of the solution:



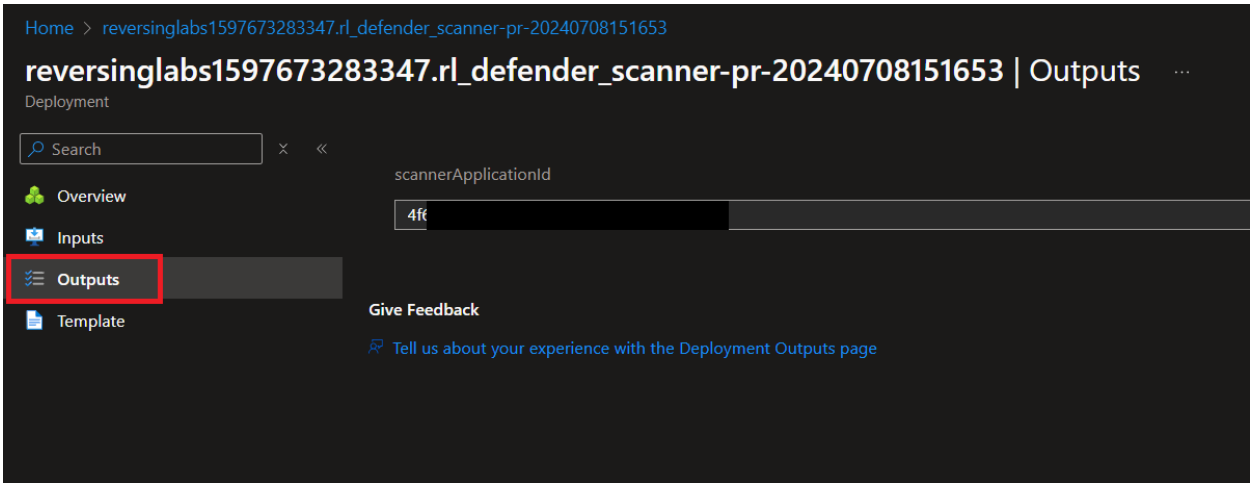
Finally, the review + create section validates the deployment template. Click “Create” to start the deployment.

2.3 Deployment Output

⚠ IMPORTANT
Ensure the steps in this section are followed. It is required to continue installation.

After the deployment is completed, an application ID value is generated that will be used in the next section. Copy this ID value.

1. Select the “Outputs” tab
2. Copy the value of the application ID:


Home > reversinglabs1597673283347.rl_defender_scanner-pr-20240708151653
reversinglabs1597673283347.rl_defender_scanner-pr-20240708151653 | Outputs
Deployment
Search
Overview
Inputs
Outputs
Template
scannerApplicationId
4f
Give Feedback
Tell us about your experience with the Deployment Outputs page

3. Graph API Permissions

An administrator needs to grant permissions to the Microsoft Graph API for the scanner to read Defender incidents.

3.1 Automated configuration

A script has been created that will automatically configure an Entra ID application with the necessary permissions. Prior to running the script, ensure that the following is installed:

- Azure CLI
- Microsoft Graph SDK for PowerShell

The steps below outline how to use the script to complete the setup process:

1. Download the PowerShell script here: DOWNLOAD
2. Run the script: powershell .\SetupRLMDEScanner.ps1
3. Paste the application ID value obtained in section 2.3:

```
C:\Users\aaron.hoffmann_rever\Dev\rl-mde-poc [develop ≡ +1 ~5 -0 !]> .\SetupRLMDEScanner.ps1

ReversingLabs Scanner for Microsoft Defender XDR Setup Script
Version: 1.0.0

[+] Checking if Azure CLI is installed...
[+] Checking if signed in to Azure CLI...
[+] Signed in as: {   "TenantId": "[redacted]",   "User": "aaron.hoffmann@reversinglabsdevelop
ment.onmicrosoft.com" }
[!] Please enter the Application ID of the scanner app: 4f6[redacted]
```

A successful run of the script looks like the following:

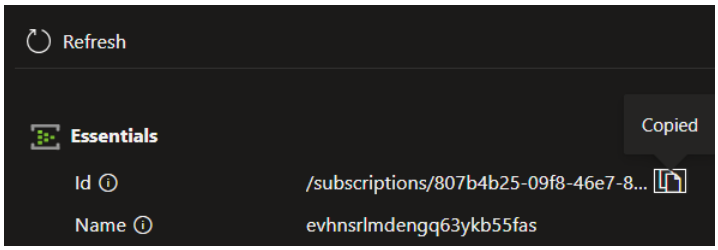
```
ReversingLabs Scanner for Microsoft Defender XDR Setup Script
Version: 1.0.0

[+] Checking if Azure CLI is installed...
[+] Checking if signed in to Azure CLI...
[+] Signed in as: {   "TenantId": "[redacted]",   "User": "aaron.hoffmann@reversinglabsdevelop
ment.onmicrosoft.com" }
[!] Please enter the Application ID of the scanner app: 4f6[redacted]
[+] Assigning permissions...
[+] Assigning scanner SecurityIncident.ReadWrite.All
[+] Assigning scanner SecurityAlert.Read.All
[+] Assigning scanner Defender ATP: Alert.Read.All
[+] Assigning scanner MS Threat Protection: Incident.ReadWrite.All
[+] Setup complete! NOTE: Microsoft Graph Permissions may take up to an hour to fully apply.
C:\Users\aaron.hoffmann_rever\Dev\rl-mde-poc [develop ≡ +1 ~5 -0 !]>
```

4. Configure the Defender Streaming API

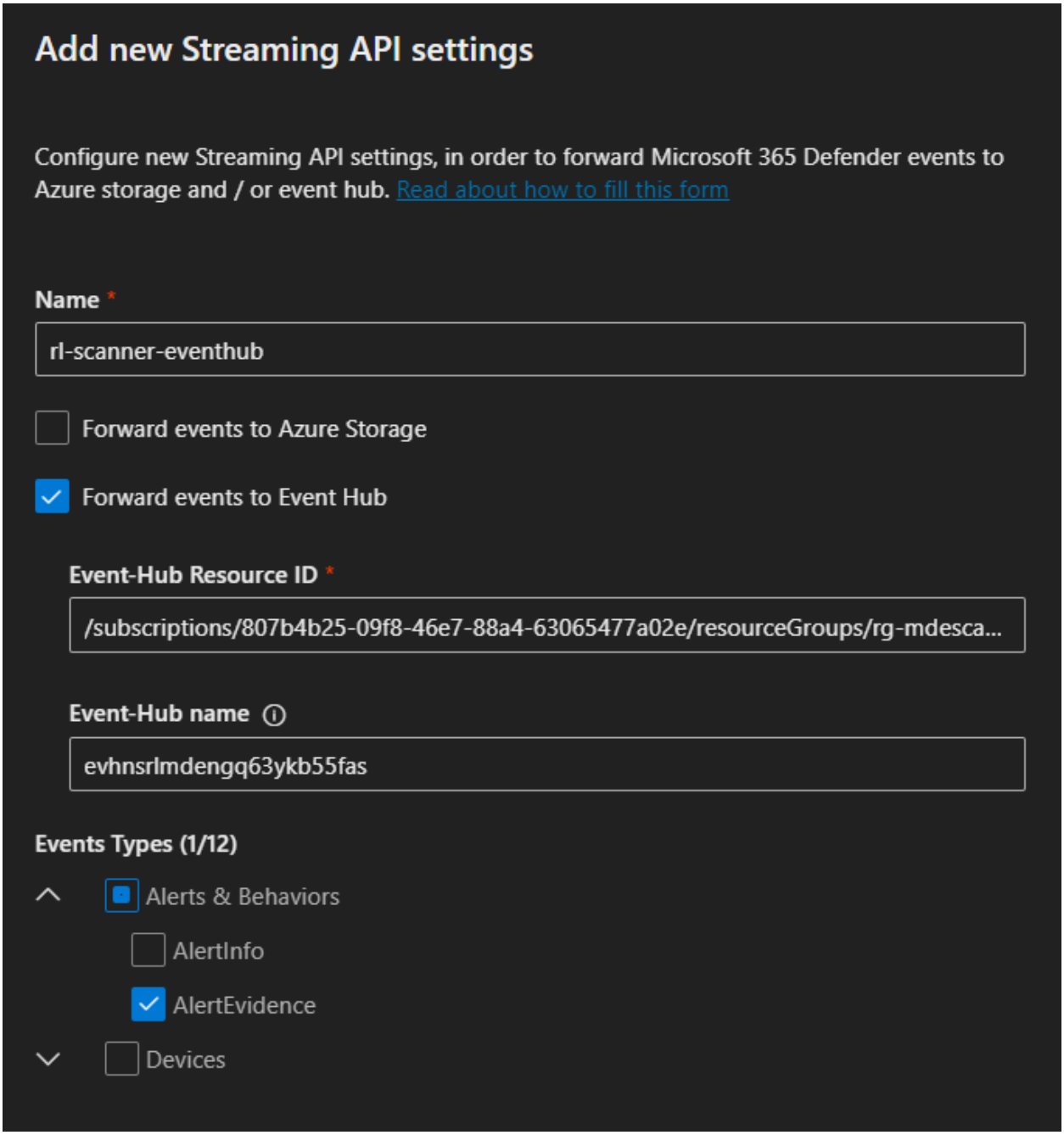
The Microsoft Defender Streaming API is the fastest way to get events from the Defender platform. The ReversingLabs scanner will read events sent to an Event Hub to identify new file hash entities for enrichment.

4.1 Setup the Streaming API

1. Navigate to the Microsoft Defender XDR portal: https://security.microsoft.com
2. Click the “System” menu, then select “Settings”
3. Select “Microsoft Defender XDR”
4. Select “Streaming API”
5. Click “Add”
6. Provide a name for the connection
7. Select “Forward events to Event Hub”
8. Obtain the Event Hub resource id and Event Hub name
    a. In the Azure portal, go to “Event Hubs” and select the event hub from the list (the name should start with “evhnsrlmde”
    b. Select “Settings”, then “Properties”
    c. Copy the “Id” value:

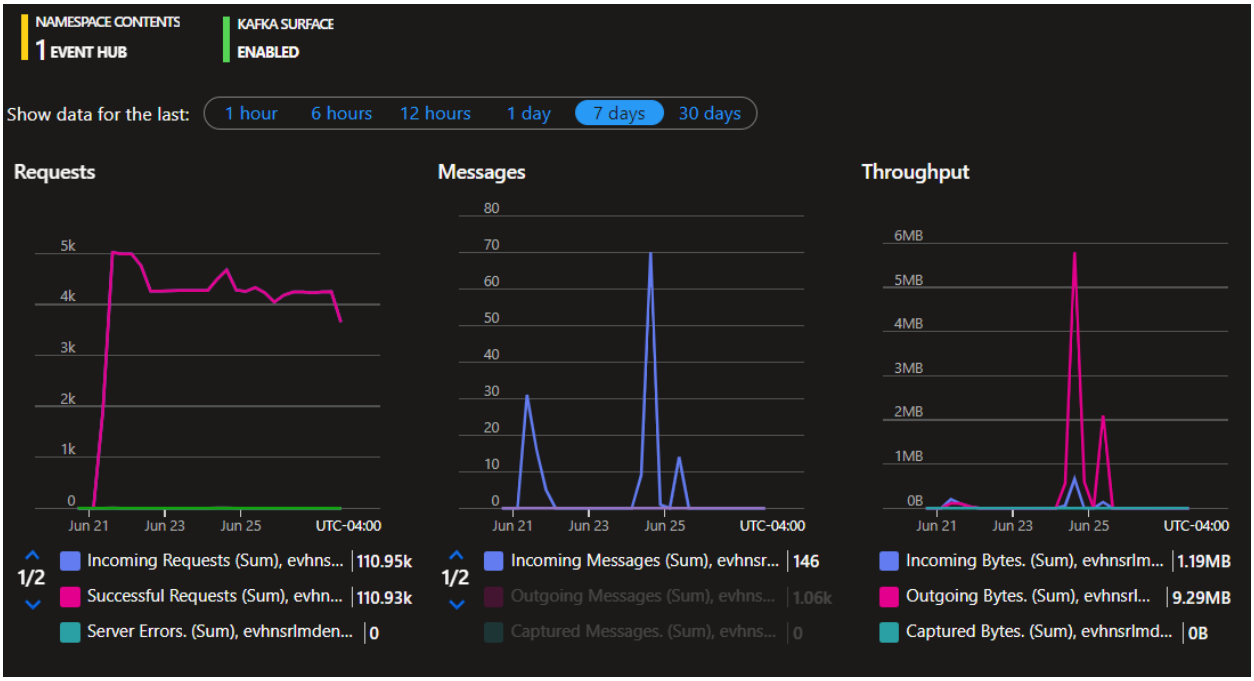


    d. The event hub name should match the name of the event hub namespace.
9. Back in the Defender portal, paste in the values.
10. Click the arrow next to the “Alerts & Behaviors” event type, then check the “AlertEvidence” box.
11. Click “submit” to enable the streaming API.



4.2 Validate Streaming API

To validate that the streaming API is working, wait for new Defender alerts or incidents to be generated, then navigate to the Event Hub resource. In the Overview menu, ensure that the “Messages” graph shows incoming messages:


NAMESPACE CONTENTS
1 EVENT HUB
KAFKA SURFACE
ENABLED
Show data for the last:
1 hour
6 hours
12 hours
1 day
7 days
30 days
Requests
5k
4k
3k
2k
1k
0
Jun 21
Jun 23
Jun 25
UTC-04:00
1/2
Incoming Requests (Sum), evhns... 110.95k
Successful Requests (Sum), evhn... 110.93k
Server Errors. (Sum), evhnsrlmden... 0
Messages
80
70
60
50
40
30
20
10
0
Jun 21
Jun 23
Jun 25
UTC-04:00
1/2
Incoming Messages (Sum), evhnsr... 146
Outgoing Messages (Sum), evhns... 1.06k
Captured Messages. (Sum), evhns... 0
Throughput
6MB
5MB
4MB
3MB
2MB
1MB
0B
Jun 21
Jun 23
Jun 25
UTC-04:00
Incoming Bytes. (Sum), evhnsrlm... 1.19MB
Outgoing Bytes. (Sum), evhnsrl... 9.29MB
Captured Bytes. (Sum), evhnsrlmd... 0B

5. Logging

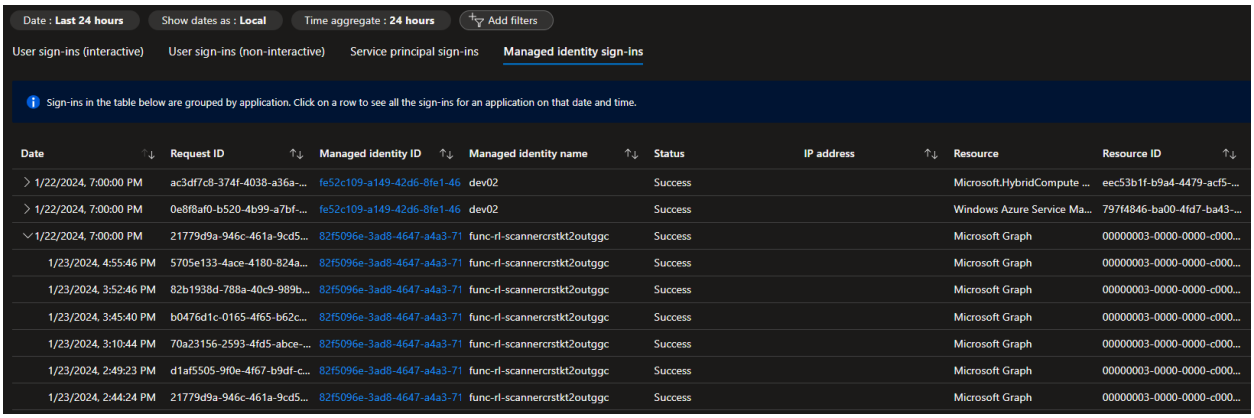
Logs from the solution are written to an Azure Log Analytics workspace. This log data is generally stored in JSON format.

5.1 Types of Logs

- **Application Logs** - these are logs produced by the application itself, e.g. events regarding file analysis.
- **Azure Resource Logs -** these are logs produced by the underlying Azure infrastructure, e.g. performance metrics.

5.2 Monitoring Recommendations

Entra ID Authentication Activity
This solution accesses Microsoft Graph and some Azure resources using a System-Assigned Managed Identity. Authentication events are logged via Entra ID under “Managed identity sign-ins”:



Date	Request ID	Managed identity ID	Managed identity name	Status	IP address	Resource	Resource ID
1/22/2024, 7:00:00 PM	ac3df7c8-374f-4038-a36a-...	fe52c109-a149-42d6-8fe1-46	dev02	Success		Microsoft.HybridCompute ...	eec53b1f-b9a4-4479-acf5-...
1/22/2024, 7:00:00 PM	0e8f8af0-b520-4b99-a7bf-...	fe52c109-a149-42d6-8fe1-46	dev02	Success		Windows Azure Service Ma...	797f4846-ba00-4fd7-ba43-...
1/22/2024, 7:00:00 PM	21779d9a-946c-461a-9cd5...	82f5096e-3ad8-4647-a4a3-71	func-rl-scannercrstkt2outggc	Success		Microsoft Graph	00000003-0000-0000-c000...
1/23/2024, 4:55:46 PM	5705e133-4ace-4180-824a...	82f5096e-3ad8-4647-a4a3-71	func-rl-scannercrstkt2outggc	Success		Microsoft Graph	00000003-0000-0000-c000...
1/23/2024, 3:52:46 PM	82b1938d-788a-40c9-989b...	82f5096e-3ad8-4647-a4a3-71	func-rl-scannercrstkt2outggc	Success		Microsoft Graph	00000003-0000-0000-c000...
1/23/2024, 3:45:40 PM	b0476d1c-0165-4f65-b62c...	82f5096e-3ad8-4647-a4a3-71	func-rl-scannercrstkt2outggc	Success		Microsoft Graph	00000003-0000-0000-c000...
1/23/2024, 3:10:44 PM	70a23156-2593-4fd5-abce-...	82f5096e-3ad8-4647-a4a3-71	func-rl-scannercrstkt2outggc	Success		Microsoft Graph	00000003-0000-0000-c000...
1/23/2024, 2:49:23 PM	d1af5505-9f0e-4f67-b9df-c...	82f5096e-3ad8-4647-a4a3-71	func-rl-scannercrstkt2outggc	Success		Microsoft Graph	00000003-0000-0000-c000...
1/23/2024, 2:44:24 PM	21779d9a-946c-461a-9cd5...	82f5096e-3ad8-4647-a4a3-71	func-rl-scannercrstkt2outggc	Success		Microsoft Graph	00000003-0000-0000-c000...

The attack surface of Managed Identities is extremely limited, however there are key items to be aware of:

- Because Managed Identity authentication occurs within Azure, this means there should be no authentication attempts from external IP addresses.
- The Managed Identity should only authenticate to the following resources:
  - Microsoft Graph
  - Azure Key Vault
  - Windows Threat Protection
  - WindowsDefenderATP

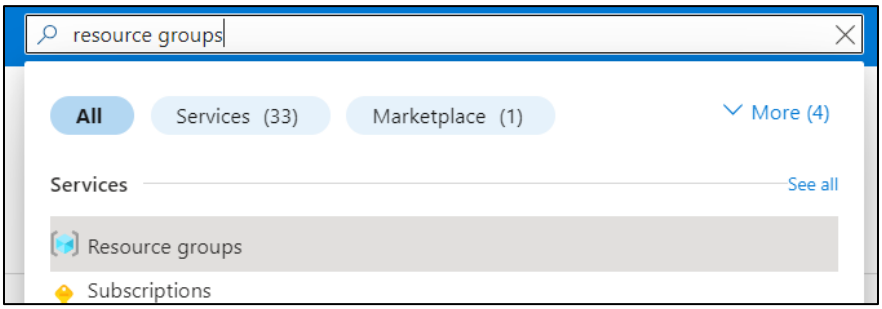
6. Support & Troubleshooting

Support-related issues should be sent to support@reversinglabs.com to create a ticket and get assistance from ReversingLabs.

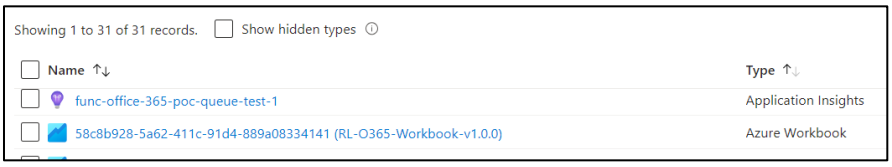
6.1 Viewing Log Data

Logs for the solution are available in an Azure Application Insights workspace. This workspace should have been automatically created during initial deployment. To access the Application Insights workspace:

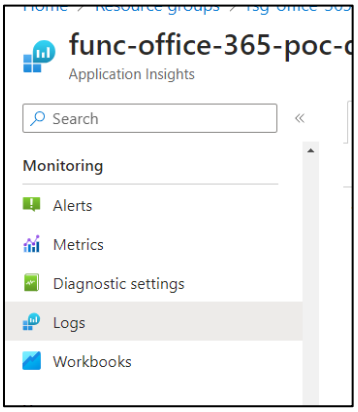
1. Log in to the Azure portal.
2. Enter and select “Resource groups” in the resource search bar:



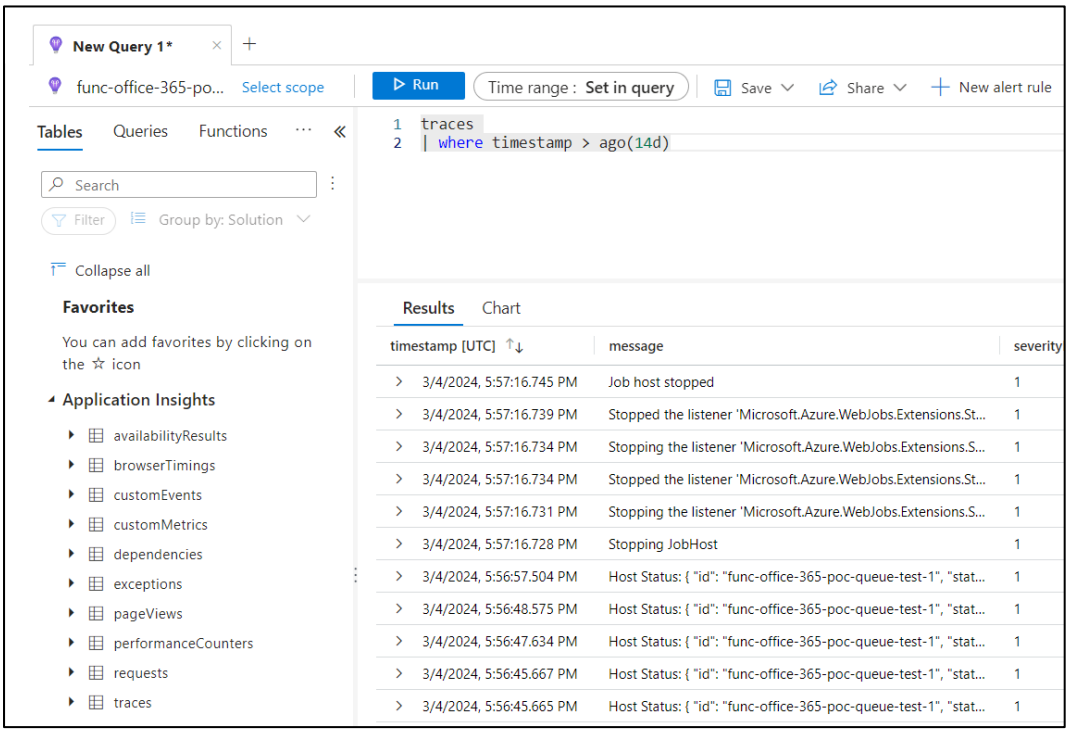
3. Search for the resource group where the solution is stored.
4. In the resources list, look for the type “Application Insights”. Click the resource name.



5. In the resource view, scroll down to the “Monitoring” section and click “Logs”:



6. Clicking the Logs item will open the query builder. Kusto Query Language (KQL) can be used to query log data for the solution. The table below provides examples.



Query	Description
`traces` `| where timestamp > ago(14d)`	This query retrieves all log data for the specified period of time. Example provided retrieves logs for the last 14 days. Use timespan values to change the lookup time period.
`exceptions` `| where timestamp > ago(14d)`	This query retrieves all errors and exceptions that were triggered over the defined period of time. Example provided retrieves logs for the last 14 days. Use timespan values to change the lookup time period.

6.2 Exporting log data

In the event log data is needed for troubleshooting, the Application Insights workspace makes it easy to export log data. Follow the steps below to export log data.

1. Use the steps outlined in the previous section to access the solution logs.

2. In the query builder, copy and paste the query below and click “Run”:

```
traces
| where timestamp > ago(14d)
| union (exceptions | where timestamp > ago(14d))
```

3. Click the “Export” button and select “CSV (all columns)”:

4. A download containing the exported logs should begin.

6.3 Disabling the solution

To temporarily disable the solution from scanning, use the following steps:

1. In the Azure resource search bar, enter “Function App”
2. Click the function app associated with the solution (Default: rlmdescanner-<random>)
3. Click the “Stop” button:

To re-enable the solution, follow the steps above. The “Stop” button will be replaced with a “Start” button.

6.4 Uninstalling the solution

The recommended method to uninstall the solution and remove all associated Azure resources is as follows:

1. Delete the resource group containing the solution resources.

7. Release Notes

2024-07-09: v0.1.0

What's New

- Beta solution finalized