


Microsoft Purview Information Protection - replacing AIP connector

Last updated by | Hekmat Abuzahrah | Mar 30, 2023 at 8:35 AM CDT

Full doc is here : <https://eng.ms/docs/security-compliance-identity-and-management-scim/modern-protection-soc-mps/sentinel/sentinel/microsoft-azure-sentinel/azure-sentinel-operational-guides/helpguides/connectors/microsoftpurviewinformationprotection/mpiphelphelpguide> 

Background

Microsoft Purview Information Protection data connector collects Office sensitivity label events and Azure Information Protection events that are published to Office. This is a Scuba routing rule based connector, similar to other Office connectors it collects the tenant audit logs through the Office Management API.

Unlike other Office-based connectors, this connector collects events from multiple Office services. Scuba polls the Office Management API and publishes to Log Analytics records based on the record type or operation name. By collecting a subset of events with a specific operation name we should automatically collect events from services that will add sensitivity label events in the future.

The following Audit logs are collected:

Record Types

- AipDiscover
- AipFileDeleted
- AipHeartBeat
- AipProtectionAction
- AipScannerDiscoverEvent
- AipSensitivityLabelAction
- SensitivityLabelAction
- SensitivityLabelPolicyMatch
- SensitivityLabeledFileAction
- MIPLabel
- MipAutoLabelSharePointItem
- MipAutoLabelSharePointPolicyLocation
- MipAutoLabelExchangeItem
- MicrosoftTeamsSensitivityLabelAction

Operations

- FileSensitivityLabelApplied
- FileSensitivityLabelChanged
- FileSensitivityLabelRemoved
- SensitivityLabelApplied

- SensitivityLabelRemoved
- SensitivityLabelRecommended
- SensitivityLabelUpdated
- SensitivityLabelChanged
- sitesensitivitylabelapplied
- sitesensitivitylabelchanged
- sitesensitivitylabelremoved
- documentsensitivitymismatchdetected

The collected record types and operations are defined in the SecEng-Scuba-Platform repo.

The Microsoft Purview Information Protection table is defined in the AM-CMS-Artifacts repository.

Tools

Scuba Office Log Collector Dashboard

Scuba is responsible for collecting Office audit logs and publishing them to Log Analytics. The following dashboard provides an overview of their Office Log Collector service.

O365 Tenant Troubleshooting Dashboard

The dashboard is split into multiple stages:

- Stage 1: ingestion - the process of polling the tenant audit logs from the Office Management API.
 - The data is pulled from Office General, SharePoint, and Exchange buckets.
- Stage 2: Processing - internal Scuba processing - filtering and batching the audit logs.
- Stage 3: Publishing - publishing the filtered events to Log Analytics.

Each stage has an overview of successes and failures.

Ingestion latency dashboards

The following dashboards are based on Log Analytics SLIM metrics that measure the Log Analytics end-to-end ingestion latency (from ODS to Kusto) per workspace per data types. They can be used to verify that the ingestion latency isn't caused by issues in the Log Analytics pipeline.

- All other regions
- EUS
- EUS2
- SUK
- WEU
- WUS2

Issues

Ingestion latency


Office Latency

Ingestion latency might be caused by the time it takes office to process the events. We're dependant on the events to be available by the Office Management API for them to be pulled and ingested into the customers workspace.


<https://learn.microsoft.com/en-us/microsoft-365/compliance/audit-log-search?view=o365-worldwide#frequently-asked-questions>

Most auditing data is available within 60-90 minutes but it may take up to 24 hours after an event occurs for the corresponding audit log entry to be displayed in the search results. See the Before you search the audit log section of this article that shows the time it takes for events in the different services to be available.

Log Analytics ingestion latency

- Use the Ingestion latency dashboard to identify ingestion issues in Log Analytics (The units are in Milliseconds). If the latency is caused by Log Analytics, open a CRI on **Azure Log Analytics/Ingestion** IcM team.  Log Analytics ingestion latency dashboard

Scuba ingestion latency

- Use the Scuba Office Log Collector dashboard to identify ingestion latency issues in Scuba. If the latency is caused by Scuba, open a CRI on **Scuba Security Platform/Platform-Sentinel** IcM team.  Scuba ingestion latency dashboard
 - Note Microsoft Purview Information Protection connector is denoted here as *MipExchange*, *MIPSharePoint*, and *MIPGeneral*.

Data is not ingested into the workspace

Possible causes

- Microsoft Purview Information Protection data connector is not connected.
 - If the issue started more than a day ago, search for a routing rule. The following query should return a Scuba routing rule snapshot record.


```
let workspaceId = <WorkspaceId>;
cluster('scubaops.westus2.kusto.windows.net')
    .database("Sentinel_Rules")
    .Rules_Office365_Snapshots_GetLatest
    | where ruleJSON contains workspaceId
```
 - The following query should return a record with a `successful` `resultType`. Change the `customerWorkspaceId` and `daysAgoConnected` variables to the appropriate values.

```

let customerWorkspaceId = <WorkspaceId>;
let daysAgoConnected = ago(14d);
cluster('securityinsights.kusto.windows.net').database('SecurityInsightsProd').Serv
| where env_time > daysAgoConnected
| where operationName == "Sentinel.Connectors.ConnectorsArmApi.Controllers.Connecto
| where toString(customData.ConnectorKind) == "MicrosoftPurviewInformationProtectio
| project
    env_time,
    resultType,
    sessionId = toString(customData["x-ms-client-session-id"][0]),
    workspaceName = toString(customData.workspaceName),
    workspaceId = toString(customData.workspaceId),
    resourceGroup = toString(customData.resourceGroupName),
    subscriptionId = toString(customData.subscriptionId),
    tenantId = toString(customData.workspaceTenantId),
    rootOperationId,
    resultSignature,
    resultDescription
| where workspaceId == customerWorkspaceId

```



- If both queries return no results, ask the users to verify the data connector is connected.
- Office audit logs aren't enabled in the tenant's account.
 - Users need to enable audit logs in the Office compliance portal.
 - <https://learn.microsoft.com/en-us/microsoft-365/compliance/audit-log-enable-disable?view=o365-worldwide#turn-on-auditing>
- No sensitivity label events data is generated in the Office Audit logs.
 - Ask the users to verify the missing events are found when they search them for them in the Office compliance portal.
- Expired license.
 - Expired license can be identified using the Scuba Office Log Collector dashboard. The "Failure - Failed API Calls" widget will have **AF20023** failures.  Scuba failed api calls dashboard
- Issues with the Scuba office log collector.
 - Other issues can be identified with Scuba Office Log Collector dashboard, if there are failures in "Failure - Internal Workload Processing Failed" widget or "Failure - Errors hit sending to OMS VIA HTTP", open a CRI on **Scuba Security Platform/Platform-Sentinel** IcM team.
 - Note that the dashboard also contains a "How to troubleshoot Stage 1(Ingestion) Issues" section and its instructions should be followed.
- If the steps didn't identify the issue, open a CRI on **Microsoft Azure Sentinel/Data Collection and Processing** team.

Missing Office Audit Logs records

First verify that the events are present in the Office Audit Logs compliance portal.

Possible Causes

Scuba

Identify possible issues in Scuba with the Scuba Office Log Collector dashboard. Check the "Failure - Failed API calls (All Products)", "Failure - Internal Workload Processing Failed", and "Failure - Errors hit sending to OMS VIA HTTP" widgets. If a problem is identified, open a CRI on **Scuba Security Platform/Platform-Sentinel** ICM team.

New Office Record Type or Operation

A new Office service added support for sensitivity label events and the events are not ingested into Sentinel. The service might have created a custom record type for those events, and we'll have to add them to the polling logic. Get audit log record samples from the customer.

- **CSS:** Open a CRI on **Microsoft Azure Sentinel/Data Collection and Processing** team.
- **ENG:**
 - Identify the missing record type or operation name and search for it the O365Exchange Nibiru repository.
 - Contact the **Scuba** team and ask them to add the record type to the *MIPGeneral* workload filtering rules.
 - Update the Log Analytics table manifest according to the interface defined in the Nibiru repository. <https://1dsdocs.azurewebsites.net/articles/loganalytics-onboarding-guide/onboarding-manifest.html>
 - See Nibiru repository section below for an example.

Missing Azure Information Protection records

Additionally to Office sensitivity label records this connector collects Azure Information Protection records that are ingested into Office Audit Logs. The Azure Information Protection data connector is deprecated and we can't guarantee 100% parity with its data.

If the user identified AIP audit logs records that we do not collect:

- **CSS:** Open a CRI on **Microsoft Azure Sentinel/Data Collection and Processing** team.
- **ENG:**
 - Identify the missing record type.
 - Search for it the [O365Exchange Nibiru repository] (https://o365exchange.visualstudio.com/O365%20Core/_git/Nibiru?path=/sources/dev/Auditing/src/Common/ComplianceAuditSchema&_a=contents&version=GBmaster)
 - Contact the **Scuba** team and ask them to add the record type to the *MIPGeneral* workload filtering rules.
 - Update the Log Analytics table manifest according to the interface defined in the Nibiru repository. <https://1dsdocs.azurewebsites.net/articles/loganalytics-onboarding-guide/onboarding-manifest.html>
 - See Nibiru repository section below for an example.

Missing columns in events

The audit logs schema has changed, or customers need columns that we do not ingest. In that instance, we'll need sample audit log records that contain the missing data.

- **CSS:** Open a CRI on **Microsoft Azure Sentinel/Data Collection and Processing** team.
- **ENG:**
 - Identify the record type or operation name that are missing columns and search for it the O365Exchange Nibiru repository
 - Update the Log Analytics table manifest according to the interface defined in the Nibiru repository. <https://1dsdocs.azurewebsites.net/articles/loganalytics-onboarding-guide/onboarding-manifest.html>
 - See Nibiru repository section below for an example.

Useful Links

- O365Exchange Nibiru repository - Contains XML files with Office audit log interfaces.
- Log Analytics manifest onboarding documentation - Instruction for writing and deploying Log Analytics tables.
- O365 Tenant Troubleshooting dashboard - Scuba Office Log Collector dashboard.
- Log Analytics ingestion latency dashboards

Nibiru repository

The Nibiru repository, contains the definition for all Office Audit Logs record types and can be used as a reference for updating the table schema and transformation.

The following Complex item defines the SensitivityLabelEventData property type interface.

```
<ComplexType Name="SensitivityLabelEventData">
  <Property Name="SensitivityLabelId" Type="Edm.String" />
  <Property Name="SensitivityLabelOwnerEmail" Type="Edm.String">
    <Annotation Term="Microsoft.Office.Audit.Schema.PIIFlag" Bool="true"/>
  </Property>
  <Property Name="OldSensitivityLabelId" Type="Edm.String" />
  <Property Name="OldSensitivityLabelOwnerEmail" Type="Edm.String">
    <Annotation Term="Microsoft.Office.Audit.Schema.PIIFlag" Bool="true"/>
  </Property>
  <Property Name="LabelEventType" Type="Self.LabelEventType"/>
  <Property Name="ActionSource" Type="Self.ActionSource"/>
  <Property Name="ActionSourceDetail" Type="Self.ActionSourceDetail"/>
  <Property Name="JustificationText" Type="Edm.String" >
    <Annotation Term="Microsoft.Office.Audit.Schema.PIIFlag" Bool="true"/>
  </Property>
  <Property Name="SensitivityLabelPolicyId" Type="Edm.String" />
</ComplexType>
```

In AipAuditSchema.xml, multiple record types use this property in their schema.

```
<EntityType Name="AipDiscover" Abstract="false" BaseType="AuditRecord">
  <Annotation Term="Microsoft.Office.Audit.Schema.RecordType" EnumMember="Microsoft.Office
  <Property Name="SensitivityLabelEventData" Type="MIP.SensitivityLabelEventData" />
  <Property Name="SensitiveInfoTypeData" Type="Collection(MIP.SensitiveInfoTypeData)" />
  <Property Name="ProtectionEventData" Type="Self.ProtectionEventData" />
  <Property Name="Common" Type="Self.AipCommon" />
  <Property Name="DataState" Type="Edm.String"/>
</EntityType>
```

Any change in the SensitivityLabelEventData property and AipDiscover record type should be represented in the Microsoft Purview Information Protection table, by updating the manifest file and KQL transformation if needed.

