AQUILA - Google Workspace Integration

Google Workspace Integration Overview

The Google Workspace integration collects and parses data from various **Google Workspace audit reports APIs** using a service account authorized via the **Admin SDK API**.

Requirements

To ingest data from the Google Reports API, the following must be completed:

- An **administrator account** in Google Workspace.
- Enable the **Admin SDK API** in GCP.
- Create and configure a **Service Account**.
- Enable **Domain-Wide Delegation** for the service account.
- Configure the **OAuth Consent Screen**.

1.Enable Admin SDK API

Our AQUILA agent will eventually use our GCP service account, which uses the Workspace Admin SDK to interact with the GW admin console REST API, therefore it needs to be enabled in GCP. To keep your mind at ease, we will only be enabling read access to the Reports API for this admin SDK.

Complete the following steps:

- Select the Google Cloud navigation menu > **APIs & Services** > **Enabled APIs & Services**
- Search and enable “**Admin SDK API**” from the **API library page**

When finished, you will have enabled the Admin SDK API within your project, where your service account will have access to pull data from GW.

2.Configure OAuth Consent Screen

We next need to set up the OAuth consent screen for our service account and application when they create API requests to GW, as it will include the necessary authorization token.

Complete the following steps:

1. Select the Google Cloud navigation menu > **APIs & Services** > **Enabled APIs & Services** > **OAuth Consent Screen**
2. User Type > Internal > Create
3. Fill out the following information in subsequent steps
4. App name:
5. User support email:
6. Authorized domains:
7. Developer contact information:
8. Save and Continue
9. Save and Continue
10. Back to Dashboard

When finished, we will now have a registered application using OAuth 2.0 for authorization and the consent screen information set. Please note, the default token request limit for this app daily is 10,000 but can be increased. We recommend setting your agent's pull rate to every 10 minutes which should not come close to this reaching this threshold. Setting the agent's pull rate will be done at a later step.

3.Create a Service Account

For the AQUILA agent to ingest data from GW, we will need to create a service account for the agent to use. This account is meant for non-human applications, allowing it to access resources in GW via the Admin SDK API we enabled earlier.

To create a service account, do the following:

1. Select the navigation menu in Google Cloud > **APIs & Services** > **Credentials** > **Create Credentials** > **Service Account**
2. Enter the following information:
3. Service account name: a
4. Service account ID:
5. Leave the rest blank and continue
6. Select your new **Service Account** > **Keys** > **Add Key** > **Create New Key** > **JSON**

By default, the Owner role will be applied to this service account based on inheritance from the project, feel free to scope permissions tighter as best seen fit. When finished, you should have a

service account, credentials for this service account in a JSON file saved to your host. We will enter this information during our GW integration setup.

4.Enable Domain-Wide Delegation

Our service account will need domain-wide delegation of permissions to access APIs that reach outside of GCP and into GW. The important data necessary for this has already been established in earlier steps where we need an API key, service account and OAuth client ID.

To enable domain-wide delegation for your service account, do the following:

1. In your GW Admin Console select > **Navigation Menu** > **Security** > **Access and data control** > **API controls**
2. Select **Manage Domain Wide Delegation** > **Add New**
3. Client ID: OAuth ID from Service Account in GCP
4. Google Cloud Console > **IAM & Admin** > **Service Accounts** > **OAuth 2 Client ID** (copy to clipboard)
5. **OAuth Scopes**: https://www.googleapis.com/auth/admin.reports.audit.readonly

Our service account in GCP only needs access to admin.reports.audit.readonly to access GW Audit Reports where these are converted into ECS documents.

If you made it this far, CONGRATULATIONS you are doing outstanding! Your GW and GCP environments are now set up and finished. At this point you are almost done.

> **Please provide the following information to CyTech Support:**

- **Delegated Account -** the email of the administrator account, and not the email of the ServiceAccount.
- **Jwt JSON** - The JSON credentials file downloaded from GCP. Raw contents of the JWT file. Useful when hosting a file along with the agent is not possible. NOTE: Please use either JWT File or JWT JSON parameter.

Reference link: https://www.elastic.co/security-labs/google-workspace-attack-surface-part-two

*If you need further assistance, kindly contact our support at **support@cytechint.com** for prompt assistance and guidance.*

Revision #1
Created 14 July 2025 09:41:30 by Richmond Abella

Updated 14 July 2025 12:22:40 by Richmond Abella