

# Google Workspace Integrations

## Introduction

Google Workspace (formerly G Suite) is a suite of cloud computing, productivity and collaboration tools, software and products developed and marketed by Google. It allows users to create, edit, and share documents, spreadsheets, presentations, and more. It also includes email, calendar, chat, and video conferencing tools.

Google Workspace is designed for businesses of all sizes, from small businesses to large enterprises. It is a popular choice for businesses because it is affordable, easy to use, and secure.

The Google Workspace integration collects and parses data from the different [Google Workspace audit reports APIs](#).

If you want to know more about how you can fully leverage the Google Workspace integration, there is a multipart blog from our Security Labs that will help you:

1. To understand what Google Workspace is in [Part One - Surveying the Land](#)
2. To set it up, step by step, in [Part Two - Setup Threat Detection with Elastic](#)
3. And to use the collected information to your advantage in [Part Three - Detecting Common Threats](#)

## Assumptions

The procedures described in Section 3 assumes that a Log Collector has already been setup.

## Compatibility

It is compatible with a subset of applications under the [Google Reports API v1](#).

## Requirements

The procedures described in Section 3 assumes that a Log Collector has already been setup.

### In order to ingest data from the Google Reports API you must:

- Have an administrator account.
- Set up a ServiceAccount using the administrator account.
- Set up access to the Admin SDK API for the ServiceAccount.
- Enable Domain-Wide Delegation for your ServiceAccount.

### Create access credentials

Credentials are used to obtain an access token from Google's authorization servers so your app can call Google Workspace APIs. This guide describes how to choose and set up the credentials your app needs.

### Choose the access credential that is right for you

The required credentials depends on the type of data, platform, and access methodology of your app. There are three types of credential types available:

- **API key credentials** - An API key is a long string containing upper and lower case letters, numbers, underscores, and hyphens, such as AlzaSyDaGmWKa4JsXZ-HjGw7ISLn\_3namBGewQe. This authentication method is used to anonymously access publicly-available data, such as Google Workspace files shared using the "Anyone on the Internet with this link" sharing setting. For more details, see [Using API keys](#). To create an API key:
  - In the Google Cloud console, go to Menu > **APIs & Services > Credentials**.
  - Click Create credentials > API key.
  - Your new API key is displayed.
    - Click Copy to copy your API key for use in your app's code. The API key can also be found in the "API keys" section of your project's credentials.
    - Click Restrict key to update advanced settings and limit use of your API key. For more details, see [Applying API key restrictions](#).
- **OAuth client ID credentials** - To authenticate as an end user and access user data in your app, you need to create one or more OAuth 2.0 Client IDs. A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms,

you must create a separate client ID for each platform. Choose your [application type](#) for specific instructions about how to create an OAuth client ID:

- Web application
  - Android
  - iOS
  - Chrome app
  - Desktop app
  - TVs & limited-input devices
  - Universal Windows Platform (UWP)
- **Service account credentials** - A service account is a special kind of account used by an application, rather than a person. You can use a service account to access data or perform actions by the robot account, or to access data on behalf of Google Workspace or Cloud Identity users. For more information, see [Understanding service accounts](#). Create a service account:

- In the Google Cloud console, go to Menu > IAM & Admin > Service Accounts.
- Click Create service account.
- Fill in the service account details, then click Create and continue.
- Note: By default, Google creates a unique service account ID. If you would like to change the ID, modify the ID in the service account ID field.
- Optional: Assign roles to your service account to grant access to your Google Cloud project's resources. For more details, refer to [Granting, changing, and revoking access to resources](#).
- Click **Continue**.
- *Optional:* Enter users or groups that can manage and perform actions with this service account. For more details, refer to [Managing service account impersonation](#).
- Click **Done**. Make a note of the email address for the service account.

### **Assign a role to a service account:**

You must assign a prebuilt or custom role to a service account by a super administrator account.

1. In the Google Admin console, go to Menu > **Account** > **Admin roles**.
2. Point to the role that you want to assign, and then click **Assign admin**.
3. Click **Assign service accounts**.

4. Enter the email address of the service account.
5. Click **Add > Assign role**.

### Create credentials for a service account:

You need to obtain credentials in the form of a public/private key pair. These credentials are used by your code to authorize service account actions within your app.

To obtain credentials for your service account:

1. In the Google Cloud console, go to Menu > **IAM & Admin > Service Accounts**.
2. Select your service account.
3. Click **Keys > Add key > Create new key**.
4. Select **JSON**, then click **Create**.

Your new public/private key pair is generated and downloaded to your machine as a new file. Save the downloaded JSON file as `credentials.json` in your working directory. This file is the only copy of this key. For information about how to store your key securely, see [Managing service account keys](#).

5. Click **Close**.

### Optional: Set up domain-wide delegation for a service account

To call APIs on behalf of users in a Google Workspace organization, your service account needs to be granted domain-wide delegation of authority in the Google Workspace Admin console by a super administrator account. For more information, see [Delegating domain-wide authority to a service account](#).

To set up domain-wide delegation of authority for a service account:

1. In the Google Cloud console, go to **Menu > IAM & Admin > Service Accounts**.
2. Select your service account.
3. Click **Show advanced settings**.

4. Under "Domain-wide delegation," find your service account's "Client ID." Click Copy to copy the client ID value to your clipboard.
5. If you have super administrator access to the relevant Google Workspace account, click **View Google Workspace Admin Console**, then sign in using a super administrator user account and continue following these steps.

If you don't have super administrator access to the relevant Google Workspace account, contact a super administrator for that account and send them your service account's Client ID and list of OAuth Scopes so they can complete the following steps in the Admin console.

- In the Google Admin console, go to Menu > **Security > Access and data control > API controls**.
- Click **Manage Domain Wide Delegation**.
- Click **Add new**.
- In the "Client ID" field, paste the client ID that you previously copied.
- In the "OAuth Scopes" field, enter a comma-delimited list of the scopes required by your application. This is the same set of scopes you defined when configuring the OAuth consent screen.
- Click **Authorize**.

**This integration will make use of the following oauth2 scope:**

- <https://www.googleapis.com/auth/admin.reports.audit.readonly>

Once you have downloaded your service account credentials as a JSON file, you are ready to set up your integration.

Click the Advanced option of Google Workspace Audit Reports. The default value of "API Host" is <https://www.googleapis.com>. The API Host will be used for collecting access\_transparency, admin, device, context\_aware\_access, drive, gcp, groups, group\_enterprise, login, rules, saml, token and user accounts logs.

NOTE: The Delegated Account value in the configuration, is expected to be the email of the administrator account, and not the email of the ServiceAccount.

# Google Workspace Integration Procedures

Please provide the following information to CyTech:

**Collect access\_transparency, admin, alert, context\_aware\_access, device, drive, gcp, groups, group\_enterprise, login, rules, saml, token and user accounts logs (input: httpjson).**

1. **Jwt File** - Specifies the path to the JWT credentials file. NOTE: Please use either JWT File or JWT JSON parameter.
2. **Jwt JSON** - 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.
3. **Delegated Account** - Delegated Account is required. Email of the admin user used to access the API.

---

Revision #3

Created 23 April 2024 13:16:52

Updated 24 April 2025 15:41:15 by John Polestico