Introducción

Este conjunto de scripts ha sido desarrollado para facilitar la gestión y protección de archivos en Google Drive de manera automatizada. Su propósito es identificar archivos con visibilidad pública, evaluar su nivel de criticidad según información aportada por sus propietarios y aplicar políticas de seguridad que reduzcan el riesgo de accesos no autorizados. Al automatizar estas tareas, el proceso se vuelve más eficiente y alineado con las mejores prácticas en seguridad de la información.

Flujo del Proceso

1. Identificación de Archivos Públicos:
   * El sistema comienza identificando los archivos públicos en Google Drive de cada usuario. Para esto, recopila información clave como:
     + Owner: El propietario del archivo.
     + Name: El nombre del archivo.
     + ID: El identificador único.
     + Visibility: El estado de visibilidad (público o privado).
   * Solo los archivos con visibilidad pública pasan a las siguientes etapas del proceso.
2. Solicitud de Clasificación:
   * Una vez identificados, los propietarios de los archivos públicos reciben un correo electrónico con un enlace a un formulario de Google Forms. Este formulario les permite responder preguntas relacionadas con la confidencialidad, criticidad y otros aspectos importantes de sus archivos. Las respuestas se almacenan automáticamente en una hoja de cálculo en Google Sheets para su análisis posterior.
3. Cálculo de Criticidad:
   * Usando las respuestas recopiladas, otro script evalúa cada archivo aplicando un sistema de puntuación predefinido. En base a los resultados, los archivos se categorizan en niveles de criticidad: baja, media, alta o crítica. Este sistema permite priorizar acciones según el nivel de riesgo asociado.
4. Gestión de Archivos Críticos o Altamente Críticos:
   * Para los archivos clasificados como de criticidad alta o crítica, el sistema actúa automáticamente:
     + Cambio de visibilidad: Si es posible, el script modifica la configuración del archivo a privado para restringir accesos no deseados.
     + Notificación al propietario:
       - Si el cambio se realiza con éxito, el propietario recibe un correo explicando que la acción fue tomada por razones de seguridad.
       - Si no es posible cambiar los permisos, se notifica al propietario solicitando que realice el cambio manualmente a privado y envíe evidencia del ajuste.
5. Comunicación Transparente:
   * En ambos casos, el propietario recibe un correo detallando el resultado de las acciones, ya sea para informar de los cambios realizados o para guiarlo sobre los pasos que debe completar.

Instrucciones para la ejecución de la aplicación

Una vez descargado el archivo challange.zip, descomprímelo en un directorio de tu elección. Al descomprimir, obtendrás los siguientes archivos:

* 1 - load\_mysql.py
* 2 - envio correos.py
* 3 - traer datos de forms.py
* 4 - calcular clasificacion.py
* 5 - load visibility.py
* 6 - lookup id files mysql forms.py
* 7 - restringir archivos publicos criticos.py
* 8 - envio correo notificacion.py
* .env (template con las variables necesarias para configurar la aplicación)

Template del archivo .env

El archivo .env se utiliza para almacenar de manera segura las credenciales y configuraciones sensibles del sistema. Esta práctica evita la exposición de información como contraseñas y claves API directamente en el código, mejorando la seguridad. Además, permite gestionar configuraciones de manera flexible para distintos entornos (desarrollo, producción), sin modificar el código fuente. Usar un archivo .env facilita la administración de credenciales y asegura una configuración adecuada en todo el proyecto.

El archivo .env debe configurarse adecuadamente para garantizar el correcto funcionamiento de los scripts. A continuación, se muestra el contenido base del archivo, que deberá ser modificado según las credenciales específicas de tu entorno:

1. Base de Datos MySQL:
   * DB\_HOST: Dirección del servidor de MySQL. Si estás trabajando en un servidor local, deja el valor como localhost.
   * DB\_USER: Nombre del usuario autorizado a acceder a la base de datos.
   * DB\_PASSWORD: Contraseña correspondiente al usuario especificado.
   * DB\_NAME: Nombre de la base de datos donde se almacenarán los datos del inventario de Google Drive.
2. Google Forms y Hojas de Cálculo:
   * SPREADSHEET\_ID: Identificador único de la hoja de cálculo de Google que contiene las respuestas del formulario.
3. Configuración de Correo Electrónico:
   * EMAIL\_USER: Correo electrónico que se utilizará para enviar notificaciones.
   * EMAIL\_PASSWORD: Contraseña de aplicación generada para la cuenta de correo. Es obligatorio usar una contraseña de aplicación si estás usando Gmail con autenticación en dos pasos.
   * EMAIL\_SMTP\_SERVER y EMAIL\_SMTP\_PORT: Servidor y puerto SMTP correspondientes al proveedor de correo (en este caso, Gmail).
4. Archivo de Credenciales de Servicio:
   * SERVICE\_ACCOUNT\_FILE\_1: Este archivo (client\_secret.json) contiene las credenciales necesarias para autenticar con OAuth2 y debe descargarse desde la consola de Google Cloud.
   * SERVICE\_ACCOUNT\_FILE\_2: Archivo JSON que representa la cuenta de servicio de Google, requerido para realizar operaciones específicas con los permisos adecuados.

Ejecución de los scripts

Luego de haber completado la totalidad de las variables de entorno con las credenciales correctamente, se puede proceder a la ejecución de los scripts en el siguiente orden:

1. load\_mysql.py: Carga datos en la base de datos MySQL.
2. envio correos.py: Envía un correo por cada archivo público al usuario con un enlace a un formulario de Google. Luego se debe esperar que el usuario complete el formulario para cada archivo.
3. traer datos de forms.py: Recoge los datos del formulario completado por el usuario.
4. calcular clasificacion.py: Procesa los datos y calcula las clasificaciones.
5. load visibility.py: Carga los resultados para su visualización.
6. lookup id files mysql forms.py: Asocia los datos del formulario con los archivos en MySQL.
7. restringir archivos publicos criticos.py: Restringe archivos públicos críticos.
8. envio correo notificacion.py: Envía una notificación final al usuario informando que el proceso ha concluido.