Handling Compromised GCP Credentials

Google Cloud Platform (GCP) credentials control access to your resources hosted on GCP. To keep your data secure, and protected from attackers, you must handle your credentials with utmost care.

You should protect all of your of GCP credentials from unintended access. These credentials include but are not limited to the following lists.

Service credentials created and managed in the Cloud Platform Console:

- **Service Account Private Keys (JSON and p12 files)**
- **API Keys**
- **OAuth2 Client ID Secrets**

User credentials you create and manage on developer workstations:

- **Google Cloud SDK Credentials** - Stored in the User Config directory reported by running the `gcloud info` command.
- **Browser Cookies** - This is browser-specific, but typically stored on the developer's workstation.

If you suspect that any of your credentials have been compromised, you should take immediate action to limit their impact on your GCP account.

Immediate steps to protect your GCP account

Revoke and reissue credentials

All credential types can be revoked and re-issued. Proceed carefully to ensure you do not suffer a service-outage as a result of revoking credentials.

In general, it is best to first generate a new credential, then push it to all services and users that need it, and finally revoke the old credential.

Replace Service Account Private Keys

Search for "Service Accounts" on the Cloud Platform Console and locate the affected service account. Create a new key for the service account, push that key to all the locations in which the old key was in use, and then delete the old key.

Regenerate API Keys

Search for "Credentials" in the Cloud Platform Console. Create a new API Key using the **Create credentials** button, configured the same as the compromised API Key. The restrictions on the API Key must match, otherwise you may suffer an outage. Push the API Key to all locations in which the old key was in use, and then delete the old key.

Reset OAuth2 Client ID Secrets

Note that changing a Client ID Secret will cause a temporary outage while the secret is rotated.

Search for "Credentials" in the Cloud Platform Console. Select the desired OAuth2 Client ID and edit it. Click on the **Reset Secret** button, and push the new secret to your application.

Remove Google Cloud SDK Credentials

A user whose Google Cloud SDK credentials have been compromised should visit accounts.google.com (https://accounts.google.com), navigate to **Connected apps & sites** (https://myaccount.google.com/permissions), and remove the Google Cloud SDK from the list of connected apps.

When you use `gcloud` command line tool again, it will automatically ask the user to authorize the application again.

This action can also be taken by a G Suite Admin on behalf of the user.

Invalidate Browser Cookies

A user who suspects their browser cookies have been compromised should immediately change their password on accounts.google.com (https://accounts.google.com). This will invalidate all existing cookies, and the user will be asked to login again at the browser.

This action can also be taken by a G Suite Admin on behalf of the user.

Look for unauthorized access and resources

After you have revoked leaked credentials, and restored your service, you should review all access to your GCP resources.

Primarily, you should examine your Activity Log in the Cloud Platform Console (search for "activity") in all affected GCP projects, and make sure that all accesses (especially related to the leaked credentials) are as expected.

Delete all unauthorized resources

Make sure that there are no unexpected resources, such as VMs, Google App Engine Apps, service accounts, Google Cloud Storage buckets, and so forth associated with the project.

Once you are satisfied that you have identified all unauthorized resources, you may choose to delete these resources immediately. This is especially important for Compute-style resources, which may be able to exfiltrate data or otherwise compromise your production systems.

Alternatively, you may opt to try to isolate unauthorized resources to allow your own forensics teams and Google Cloud Support (https://cloud.google.com/support/) to perform additional forensics.

Contact Google Cloud Support

Contact Google Cloud Support (https://cloud.google.com/support/).

General best practices

Separate credentials from code

Manage and store your credentials separately from your source code. It is extremely common to accidentally push both credentials and source code to a source management site like GitHub, which makes your credentials vulnerable to attack.

Service account best practices

Follow the best practices for service accounts (https://cloud.google.com/iam/docs/understanding-service-accounts).

Last updated September 3, 2019.