Separation of duties

Separation of duties is the concept of ensuring that one individual does not have all necessary permissions to be able to complete a malicious action. In Cloud Key Management Service, this could be an action such as using a key to access and decrypt data which that user should not normally have access to.

Separation of duties is a business control typically used in larger organizations, meant to help avoid security or privacy incidents and errors. It is considered best practice.

For further guidance, see our documentation on using Cloud Identity and Access Management securely (https://cloud.google.com/iam/docs/using-iam-securely).

Setting up Cloud KMS in a separate project

Cloud KMS could be run in an existing project, for example `your-project`, and this might be sensible if the data being encrypted with keys in Cloud KMS is stored in the same project.

However, any user with `owner` access on that project is then also able to manage (and perform cryptographic operations with) keys in Cloud KMS in that project. This is because the keys themselves are owned by the project, of which the user is an `owner`.

Instead, to allow for a separation of duties, you could run Cloud KMS in its own project, for example `your-key-project`. Then, depending on the strictness of your separation requirements, you could either:

- (**recommended**) Create `your-key-project` without an `owner` at the project-level, and instead have an Organization Admin granted at the organization-level (https://cloud.google.com/resource-manager/docs/quickstart#grant_iam_roles_at_the_organization_level). This Organization Admin is not like an `owner` - they are not able to manage or use keys, but they have the permission to Set Cloud IAM Policy to restrict who is given permissions for key management and use. Using an organization-level node, you can further restrict permissions for projects in your organization. See the next section for alternative roles to `owner` for use with Cloud KMS.

- (not recommended) Grant an `owner` role for `your-key-project` to a different user than the `owner` on `your-project`, where the keys from Cloud KMS are being used. This user would still have access to all key operations.

Choosing the right Cloud IAM roles

For smaller organizations, the primitive roles of `owner`, `editor` and `viewer` likely provide sufficient granularity for key management. In larger organizations where separation of duties is required, the `owner` role provides excessive access to sensitive security operations, and may not be ideal. In this case, we instead recommend you use predefined roles:

- **For the business owner whose application requires encryption**: Use the organization-level (https://cloud.google.com/resource-manager/docs/quickstart#grant_iam_roles_at_the_organization_level) Organization Admin role.
- **For the user managing Cloud KMS, that is, a member of an organization's IT security team**: The two roles with the minimum permissions required to manage Cloud KMS via the Cloud Console are the predefined Project Editor (`editor`) role, or a custom role (https://cloud.google.com/iam/docs/understanding-custom-roles) based on the Cloud KMS Admin (`cloudkms.admin`) role combined with the following permissions:
  - `serviceusage.quotas.get`
  - `serviceusage.services.get`
  - `resourcemanager.projects.get`

  For information about creating a custom role, see Creating a custom role (https://cloud.google.com/iam/docs/creating-custom-roles#creating_a_custom_role). If the user manages Cloud KMS using only the `gcloud` tool and the Cloud KMS API, the predefined Cloud KMS Admin role has sufficient permission without requiring the `serviceusage.quotas.get`, `serviceusage.services.get`, or `resourcemanager.projects.get` permissions.
- **For the user or service using keys for encryption and decryption operations**: Use a Cloud KMS project-level service account with Encrypter/Decrypter (`cloudkms.cryptoKeyEncrypterDecrypter`). If the service only needs to be able to write or only be able to read data, further restrict use with the role of only Encrypter (`cloudkms.cryptoKeyEncrypter`) or Decrypter (`cloudkms.cryptoKeyDecrypter`).

Using a separate project for Cloud KMS and these recommended roles, you address several security concerns about separation of duties:

- Any user who needs a primitive role at the project-level in order to use another Google Cloud service cannot use that authority to access Cloud KMS.
- No user will need the `owner` role, which could grant them excessive permissions to use keys. With no `owner`, no user can both manage and use a key to encrypt or decrypt data directly by default. Note that if data access logs (https://cloud.google.com/kms/docs/logging) are not enabled, an `owner` could have potentially unaudited access to use key material.

Note: using the `cloudkms.admin` role instead of the `owner` role is a deterrent to the user using keys to encrypt and decrypt data. It is not a strict constraint, and does **not** prevent a purposeful, malicious individual from using a key for encryption or decryption. They could still grant themselves the Cloud IAM permission to use the key, and then do so.

Last updated December 17, 2019.