netskope

WHITE PAPER

Securing GCP: Top Ten Mistakes to Avoid

The Google Cloud Platform (GCP) is a fast-growing, innovative public cloud solution that enables and accelerates business transformation. While it provides tremendous benefit with greater business agility, scalability and elasticity, securing data and workloads in this dynamic infrastructure remains a top challenge. Fortunately, there are steps you can take to keep your organization secure while utilizing GCP. This white paper highlights how Security Operations (SecOps) and Cloud Operations (CloudOps) administrators can address the top 10 mistakes made by organizations when adopting and securing GCP.

Like all public cloud provider solutions, GCP manages security of the cloud, while security in the cloud is the responsibility of the customer organization. Most SecOps administrators aren't aware of the best security practices and controls to follow. How do you efficiently move to GCP without increasing the risk of a security breach or data loss? How can you best assess your GCP environment for misconfigurations.

Netskope, the established market leader in cloud security solutions, helps the world's largest organizations take advantage of cloud environments while enhancing security. Using Netskope for GCP, organizations gain continuous visibility into security and compliance risks across their GCP environment and can quickly identify threats and misconfigurations and then mitigate them before they cause damage or data loss. Here are the top 10 security mistakes SecOps and CloudOps Administrators typically make and how to avoid them.

#1 NOT ENFORCING 2 FACTOR AUTHENTICATION

A password dump, phishing attack, or easily guessed password can lead to an employee leaking access to a GCP Project with their G Suite credentials. Don't leave access to GCP resources to just a password, implement two-factor authentication (2FA). Combat this is in a few different ways:

- Make sure all of your GCP users belong to the G Suite domains you administer, so you can effectively enforce 2FA.
- From each of your G Suite domains follow the recommended Google instructions to enable and enforce 2FA (G Suite Admin Help for 2-step Verification). Since each environment has different user support requirements, a best practice of having enforced 2FA of "Any" method, with a short enrollment grace period, while still allowing for remote social engineering and access, is a superior approach to using passwords only.
- For next level protection to prevent G Suite and GCP Organization takeover, create admin only user account(s) and remove admin privileges from all "normal" users. For instructions on how to setup the physical key enforcement go to Google's documentation titled: "Securing Your Account with Security Keys".

This will allow a balanced combination of more stringent 2FA for more privileged user accounts and more forgiving 2FA rules for lower risk user accounts. Permissions of note to protect with more stringent 2FA enforcement are the G Suite Super Admin role user(s), Google Domains linked admin user(s), and any Google Groups that have GCP Organization owner or Admin roles.

With Netskope, administrators can easily monitor the 2FA status of their Google G Suite users. Using Netskope and a single-sign-on (SSO) solution such as Okta, step-up authentication based on risk attributes can be established. Netskope for Google G Suite provides granular visibility and control of all G Suite services, along with the cloud services that make up the G Suite ecosystem.

#2 OVERLOOKING IAM INHERITED PERMISSIONS

In GCP, the overarching structure is based on an Organization, which usually corresponds to a domain name, like netskope.com. Within the Organization, there are Folders, which contain either other Folders or Projects. Folders can be used as Organizational Units (OUs) to represent certain business units. Projects actually contain resources, such as databases and virtual machines. Users can have access to the Organization, or just some parts of it.

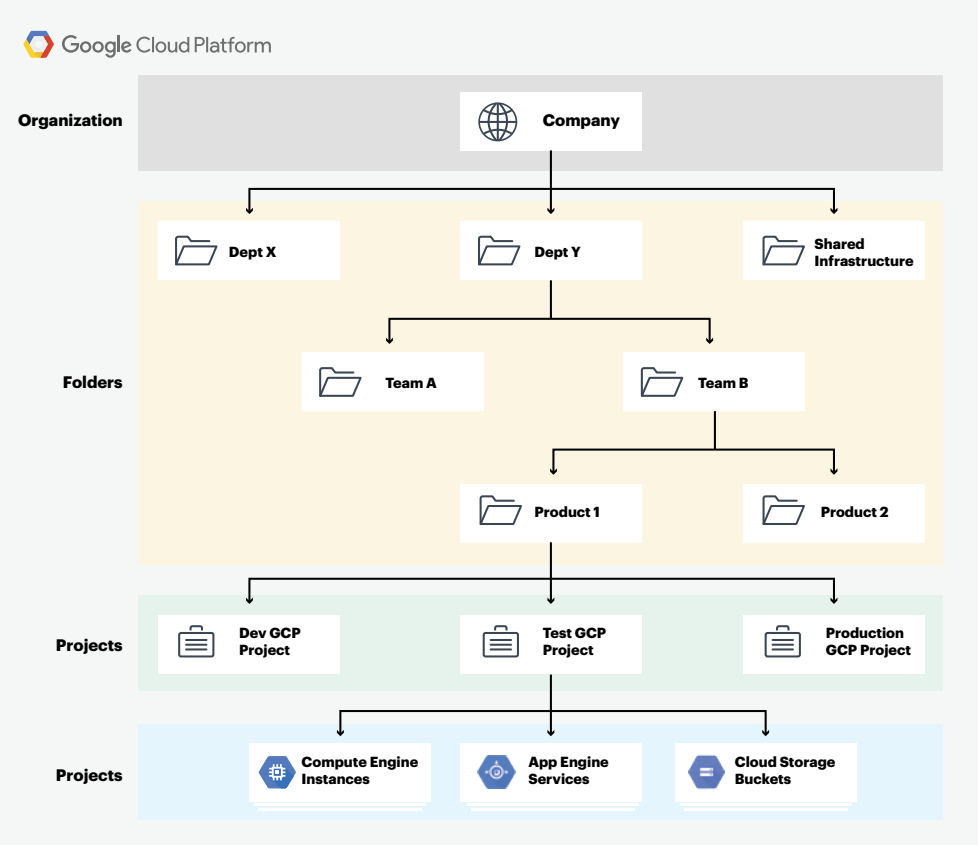

Figure 1: The GCP Resource Hierarchy

The cloud resource manager is the root of the GCP hierarchy, and is key to centralized management of your Identity and Access Management (IAM) resources. Misconfiguration here (such as giving a user 'Project Owner' permissions) could cause you to lose control of projects in your organization, as it will be automatically inherited throughout the organization. This would also grant a user the ability to create their own projects within your GCP organizational space, and they will be automatically granted the owner primitive on that new project. Some best practices include:

- Proper configuration is site specific, to learn more about about this visit the IAM domain organization root from the GCP Console, and assess all organization/sub-organization assigned permissions, as these will be inherited by all projects that exist under this GCP Organization, and in particular primitives and admin roles.

G Suite Admin Help for 2-step Verification: https://support.google.com/a/answer/9176657?hl=en#.
Securing Your Account with Security Keys: https://cloud.google.com/solutions/securing-gcp-account-security-keys#set_up_security_keys.

- Assign/find the G Suite Google Groups with admin enforced 2FA security keys. SecOps Admins may want to require physical control of security keys in a company vault or safe deposit box, as organization owner primitive, and GCP Admin roles ripple downward and potentially grant access or compromise of all managed GCP resources.

As a reminder, architect permissions well and utilize security solutions like Netskope to obtain a better understanding of what resources and services users can access.

#3 USING OUTDATED ADMINISTRATIVE ROLES

Addressing this common misconfiguration can help minimize risk of data loss. GCP contains 'Primitive Roles', which are legacy items from before they created Cloud IAM. These roles are 'Owner', 'Editor', and 'Viewer'. These each contain rather broad permissions that are not recommended for use.

To prove the point that primitives are bad, Editor primitives have a unique default authorization in Google Kubernetes Engine (GKE). They get the equivalent GKE functionality of the 'container.admin' role. This permission would allow any Project Editors in that project the ability to execute something in your GKE clusters.

As a best practice, use more granular permissions for your users instead of primitive roles, so your users are given the least privileges necessary to complete their tasks. Finally, watch and alert on primitive creation and existence in your GCP environment, especially at the organizational root, as mentioned in the previous topic, to prevent unintended primitive use and inheritance.

With Netskope, admins can get alerted to users or service accounts that have primitive roles assigned to them. Netskope for GCP looks for these misconfigurations at all levels of the Organization hierarchy.

#4 UNAUTHORIZED NETWORKS ACCESS IN GKE

GKE is a great solution to quickly utilize Kubernetes for cluster management and orchestration. However, when using GKE, the master nodes are not in an administrator's GCP Compute Network. The master nodes are hosted in a Google owned project network and they connect to an organization's worker nodes via VPC Network Peering. By default, the master nodes are publicly accessible.

This leads to a common misconfiguration for GKE users: No Controls from which GKE admins can connect to their GKE master nodes.

Best practices include:

- Create an Authorized Network for GKE Master access, per the documented Google recommendations "Adding authorized networks for cluster master access".
- Please note the following limitations when implementing. Whitelisted source Classless Inter-Domain Routing (CIDRs) can be no bigger that /24, and there can be no more than 20 CIDR references. This limitation may lead admins to pursue the use of bastions for greater network access control/multi-factor authentication/auditing.

Page 4: Adding authorized networks for cluster master access https://cloud.google.com/kubernetes-engine/docs/how-to/authorized-networks.

- As a reminder, don't use the 'Editor' primitive to grant access to your master nodes, and frugally grant assignment to the 'container.admins' role for authorization. Watch your configuration and alert on assignment of this role assignment, or membership outside of the valid membership lifespan if using an auto-expiring role assignment solution. This will go a long way for all GKE users, until Kubernetes Role-based Access and Control (RBAC) is implemented and tuned, which is not manageable via the GKE APIs.

Using Netskope for GCP, administrators get alerts on container admin roles being granted to users or service accounts. Additionally, they get alerted on editor permissions given for master nodes.

#5 USING DEFAULT NETWORKS AND RULES

By default, when you create a Project, a default network and some default network rules are created for all regions and availability zones. Best practice recommendations include:

- Remove unused region compute networks, from all of your Projects. Remove all the default network rules allowing SSH, RDP, and ICMP from the Internet. This enforces explicit network security rules.
- Do not delete the rule gke-<cluster_name>-<random-characters>, as this is needed for the GKLE master to node SSH communication. GKE leverages this firewall rule and network metadata to perform SSH access.
- Use compute labels with network firewall rules for internal compute communications, to further enforce least permissive access. If network communication is from or to outside Project networks, use CIDR whitelisting instead of labels.
- Monitor configurations and set them to alert on any default network, default rule existence or overly permissive broadly scoped access rules (i.e. no labels, CIDRs too big, or any/any access for multiple protocols or ports). Remove them promptly.

Netskope for GCP alerts administrators to sensitive network access, such as via SSH or RDP being open to the Internet.

#6 UNWARRANTED REMOTE ACCESS

GCP metadata is a very powerful and useful feature for controlling remote access to your instances via SSH or virtual serial console. There are several missteps to be aware of to prevent more broad access than intended, or leaked secrets.

Anyone who has used orchestration to distribute SSH keys can benefit from this tip, but stay clear from secrets storage. Secrets belong in great places, like Kubernetes secrets (if encrypted by HSM/KMS), or HashiCorp Vault, but not on the metadata side in plain text.

Best practices include:

- Configure your instances to ignore Project-wide public SSH keys, with exception of your GKE compute nodes. Refer to Google documentation on "Adding or removing project-wide public SSH keys" for more details.
- Then add public keys at the instance metadata level, as referenced in the above Google documentation.

Setting Network Policies in Kubernetes https://kubernetes.io/docs/concepts/services-networking/network-policies/

The reason GKE compute nodes should be excluded from the project-wide block is that GKE registers its public keys at the Project level. Blocking it will prevent the GKE master from being able to connect to your nodes via SSH for cluster communication, unless you write your own automation to replicate GKE SSH publickeys at the Project level to the instance level on GKE cluster nodes.

Additionally, don't activate the virtual serial console at the Project level. This should be detected, alerted, and if possible, auto-remediated. This allows public access to a compute instance, if the SSH keys, and Project are known, with no actual VPC network communication—so firewall rules will not block this. If absolutely needed, do so at the instance level and immediately disable this after you are done, per Google recommendations regarding "Enabling access for a VM instance".

Since you are using labels for firewall authorizations, you need to be watching your instance metadata for firewall referenced label reconfiguration/addition/removal, that could be changing the scope of an instances network access.

With Netskope, admins are alerted when an organization policy is set to allow virtual serial console at the Project level. Also, administrators are alerted when labels are changed on instances, which will expose sensitive ports to the Internet.

#7 ALLOWING OPEN TRAFFIC COMMUNICATION WITH CLUSTERS

Google Kubernetes Engine (GKE) supports network policies for Kubernetes clusters, but these are not enabled by default. Network policies are like firewall rules for your pods, so you can specify what IP address ranges and what namespaces can communicate with your pods. By default, when you create a cluster in GKE, the pods are not isolated and will accept network traffic from any source. You must either create a cluster with a network policy or update your clusters to use a network policy.

This is a best practice recommendation, to provide greater control over the communications with the applications running in Kubernetes. Once you have enabled network policies, you will want to configure them to your specifications. More information on how to configure network policies are available in the Kubernetes documentation online: Setting Network Policies in Kubernetes.

Netskope for GCP helps administrators ensure that network policies are enabled for their clusters and quickly alert them when changes occur.

#8 USING PUBLIC IP ADDRESSES FOR COMPUTE ENGINES

Many clients using GCP still default to public IP assignment to all compute instances. As a best practice, run services on private IPs, and only expose services via public hardened gateways, firewalls, load balancers, Web Application Firewalls (WAFs) or Content Delivery Networks (CDNs) that can log and restrict access where needed. Stay away from public IPs on your instances wherever possible, to avoid inadvertently exposing your instances via misconfigured firewall rules.

Any instances created with public IPs should be alerted for validation, as they increase external exposure and risk.

With Netskope, you are alerted on instances that are assigned public IP addresses. In the future, this would grow to include more high fidelity alerting on these, to understand which ones also have a route to the Internet and an active Internet Gateway.

#9 NOT MONITORING APIS FOR ENABLED SERVICES

If you aren't using all GCP services, you're probably in the majority of users. This is one of the final mistakes. Don't simply monitor and alert on what services you're using. Your organization could face a large bill for the malicious use of your cloud resources. Some best practices include:

- Disable the GCP APIs for unused services, until you need them.
- Watch and alert/auto-remediate if they are enabled.
- Monitor the whole GCP space.

Netskope for GCP provides Whitelist APIs as needed for select cloud services within the organization, and allows admins to stay alerted if additional services are enabled.

#10 RELYING ON DEFAULT ENCRYPTION TO PROTECT DATA

Whether it's from misconfigured authorization, inadvertent public sharing, or enabling debug in production, there are many ways to expose your sensitive data. Keeping data encrypted with keys you control is a great way to mitigate these risks.

GCP provides a Key Management Service (KMS) that allows customers to provision their own keys for encrypt /decrypt operations, called customer managed encryption keys (CMEKs). While GCP provides encryption by default for any data at rest, we recommend maintaining control over your secrets by leveraging CMEKs (or customer supplied keys where it's supported).

KMS allows customers to easily assign fine-grained permissions to who can find and use the keys. This can also facilitate/enable data labeling efforts, and add additional outlier visibility by data key usage monitoring/alerting.

With Netskope, be alerted on entities not using CMEKs that support it for encryption. Know when permissions for these keys are changed, so alerts are provided. Soon, Netskope will enable you to see anomalous usage of keys, so you can find data leaks more easily whether it's coming from inside or outside your organization.

SUMMARY

Comprehensive security for the Google Cloud Platform requires continuous visibility and control of the many services within the platform. It can be an overwhelming task for organizations both new to cloud adoption, and experienced with cloud environments. Netskope can help.

Netskope delivers 360° data protection, advanced threat protection and real-time controls, all from a cloud-native platform designed to secure IaaS environments like GCP, as well as SaaS and Web-based infrastructures. By providing almost 90 configuration checks based on GCP and Center for Internet Security (CIS) best practices, Netskope for GCP helps simplify and expedite your cloud security efforts.

Learn more about how you can boost your GCP security posture with Netskope at www.netskope.com/products/netskope-for-google-cloud-platform