W&B Security Review

SpringML

Introduction

Overview

Weights and Biases (W&B) requested Google and SpringML to conduct an initial short assessment of their Google Cloud instance. The assessment is focused on a preliminary review of the Google Cloud instance within W&B focusing on the following areas:

- IAM
- Load Balancing
- Networking Rules and Configuration
- Cloud Functions
- All data storage, including Google Cloud Storage, BigQuery, and CloudSQL
- API keys
- MFA and account verification at the organization level
- Cloud Logging
- Google Kubernetes Engine

The assessment's goal was to identify gaps in the implementation of best practices for the Google Cloud instance. The report focuses on identifying the risk and impact of those gaps, as well as suggestions for implementing the suggested best practices.

This document reviews the project scope, a brief description of the systems studied, our findings and their associated suggested mitigations and exposures, and a summary of what the next steps would be.

Scope

This security review is the result of a two week assessment of the Weights & Biases GCP projects: Sandbox, Beta, QA, and Production. The focus was on IAM, general security, identifying threats and vulnerabilities, and understanding how to manage threats via built-in GCP tools.

System Description

In general, the public facing application(s) are in the Production project, with a few exceptions. The beta project allows for public facing use, but routes to resources in the production project. There are also public facing resources in the sandbox project. Typically, the web application is a Google Kubernetes Engine (GKE) cluster that accesses CloudSQL, BigQuery (BQ), Redis, and Google Cloud Storage (GCS). Cloud Functions are also utilized to access data for the web application.

For the enterprise SaaS, customers share resources: cluster, buckets, databases, etc. For other customers, a single tenant architecture is used, where each customer has their own cluster, bucket, database, etc.

Privileged customer data is stored in GCS, BQ, and CloudSQL.

Findings

The focus of the review is on IAM, identifying potential threats and vulnerabilities, and understanding how to better manage threats via built-in GCP tools.

We examined the following areas of the projects, with special attention to anything customer facing or in production:

- IAM
- Load Balancing
- Networking Rules and Configuration
- Cloud Functions
- All data storage, including Google Cloud Storage, BigQuery, and CloudSQL
- API keys
- MFA and account verification at the organization level
- Cloud Logging
- Google Kubernetes Engine

These areas comprise the most common security concerns. While some findings help reduce human errors rather than reducing the likelihood of an external attack, we thought it helpful to include them as well. Most, if not all, findings that reduce human errors also help to reduce the attack surface, or overall possible points that someone could gain malicious access. This way we were able to address software and data integrity as a whole.

Our findings are identified as vulnerabilities, and each one is given a description and a brief explanation of the mitigation that could be taken. In some cases, there are multiple options for mitigation, depending on the customer's preference, and not all options are listed.

Following the descriptions and mitigations, the likelihoods and impacts involved with the vulnerabilities are in a separate table, allowing us to estimate the potential exposure involved.

Below is the table of all vulnerabilities that were found.

Vulnerability Name	Description	Mitigation
Multiple Project Owners and Assigned Primitive Roles	Projects have multiple owners and editors.	Replace highly-privileged Owner and Editor roles with more specific roles based on services used. If the projects involve teams, assign roles to Google Groups to allow team membership to be managed separately from the technical decisions about which roles are needed. The “Project IAM Admin” role is a good one to know to allow team leads to adjust roles as needed without being Owners or Editors. IAM Activity Audit Logs would then allow you to review any particular privileges these users enable for themselves or others.
Temporary Elevated Permissions	Individual users have been assigned temporary permissions in production.	Create a Google Group for any users who might need elevated production permissions. Relegate production permissions to individuals who will consistently require them. If temporary permissions are unavoidable, use an automated method to assign them (see https://cloud.google.com/iam/docs/configuring-temporary-access).
Inactive Service Accounts	Service accounts exist and have production permissions that may not be in use any more. Determine which accounts are not in use and remove them. Limit creation of new service accounts.	Determine which accounts are not in use and remove them. Limit creation of new service accounts.
Inactive Service Accounts	Service accounts exist and have production permissions that may not be in use any more.	Determine which accounts are not in use and remove them. Limit creation of new service accounts.
User Managed Service Account Keys	Service accounts exist in production with user managed keys attached.	Remove user managed service account keys. These keys are often created so a developer can impersonate the elevated permissions a service account usually has. A service account does not usually need user managed keys. If they are needed, set a time expiration and rotate them periodically.
Elevated Organization Level Permissions	The organization has many users with elevated access.	Reduce access to organization level administration to a very small (2-3) group of people that need access to project creation, billing, and other organization level administrative tasks. Remove organization-wide developer access, assigning developer access on a project by project basis. Consider defining users & teams and the initial roles they will need to be a standard step of architecting a new project.
Elevated User Permissions in Production	Users that may not need production access have elevated permissions. Users have edit access to resources in production to GCS, GKE, BQ, CloudSQL, and other areas.	Developers and other individuals that are not tasked with fixing production bugs have elevated permissions in production and may accidentally modify production resources. It also allows a greater surface area for attacks via compromised accounts.
Manual deployment of processes in production	Allowing for manual deployment of new versions of software allows for human error in the production environment	Automate deployment processes, allowing only service accounts to execute deployments.

Vulnerability Name	Description	Mitigation
Production Resources Outside The Production Project	Production resources exist in the sandbox and beta projects where users may have more elevated permissions, and the project may not be as secure as production. This also makes it difficult to isolate and secure the production environment	Migrate all production resources into the production project.
Cloud Functions All Operate Under Default Service Account	All Cloud Functions operate under a single service account	Every critical Cloud Function should have its own service account so the principle of least privilege can be applied to each one. Non-critical (internal) Functions could be grouped under a single service account. The default service account (App Engine service account) may have permissions that are too broad.
App Engine Service Account Has Elevated Production Permissions	The default App Engine service account has broad permissions in production that may not be necessary	The App Engine service account has the following roles: Editor, Service Account Token Creator, and Storage Object Admin. Primitive roles such as Editor should be avoided. If this service account is compromised, Service Account Token Creator could be leveraged to gain access to any service account that already exists.
Cloud Armor Rules Lacks Protection Against SQLi and XSS	SQL injection and XSS (cross site scripting) are common ways to gain malicious access.	Cloud Armor provides built-in ways to tune WAF rules from common open source standards.
Unrestricted API keys	API keys exist that have no restrictions on who can use them Add restrictions to existing API keys.	Add restrictions to existing API keys.
Production CloudSQL instances allow connections without SSL	CloudSQL instances allow connections without an SSL certificate	Restrict CloudSQL connections without an SSL certificate.
Production Compute Engine Instances With Public IP Addresses	Compute Engine instances have public IP addresses that are open to the internet	Restrict all public IP addresses, enforceable at the organization level. Alternatively if public IPs are needed, consider using bastion machines to separate public IPs and limit the attack surface. Secure load balancer policies could also be used to limit the likelihood of an attack on a public facing IP address.
Open Firewall Rules	Firewall rules allow connections from all IP addresses, including on ports 20 (SSH), 80 (HTTP)	Create a custom VPC with custom firewall rules.

Table 1. Findings

When trying to identify the exposure of a vulnerability, we want to use the likelihood of a compromise along with the impact a compromise would have. If they are both low, the exposure is low, if they are both high, the exposure is high.

Liklihood	Imact		
	Low	Moderate	High
High	Low	Moderate	
Moderate	Low	Moderate	Moderate
Low	Low	Low	Low

Table 2. Risk Exposure

Name	Lilkihood	Impact	Risk Exposure
Multiple Project Owners and Assigned Primitive Roles	Low	Moderate	High
Temporary Elevated Permissions	Low	Moderate	
Inactive Service Accounts	Low	Moderate	Moderate
User Managed Service Account Keys	Low	Low	Low
Elevated Organization Level Permissions	High	High	High
Elevated User Permissions in Production	High	High	High
Manual deployment of processes in production	High	High	High
Production Resources Outside The Production Project	Moderate	High	Moderate
Cloud Functions All Operate Under Default Service Account	Low	Moderate	Moderate
App Engine Service Account Has Elevated Production Permissions	Low	High	Low
Cloud Armor Rules Lacks Protection Against SQLi and XSS	Moderate	High	Moderate
Unrestricted API keys	Low	Low	Low
Production CloudSQL instances allow connections without SSL	Moderate	Moderate	Moderate
Production Compute Engine Instances With Public IP Addresses	Moderate	Moderate	Moderate
Open Firewall Rules	Moderate	High	Moderate

Table 3. Risk Analysis

Next Steps

SpringML will review the gaps and vulnerabilities identified in this report with W&B and prioritize the implementation of suggested best practices based on the needs of the W&B team.

As indicated above, these are preliminary findings. We would like to suggest a detailed approach in addressing the high risk exposure areas as the beginning of a service offering engagement that SpringML would be glad to undertake. We would need to work closely with the W&B team to identify their priorities and work with existing deadlines on their team.

Based on initial analysis, this more detailed approach, and the vulnerabilities in this report can be addressed within the short time frame of an 8 - 10 week engagement with SpringML.