Google Cloud

Guide to log queries, exports, and analysis

When responding to a security incident, reviewing logs is essential in developing a comprehensive picture of what happened. As a cloud security analyst, you'll frequently use logs to reconstruct the actions that occurred before and after a security incident. In this reading, you'll explore Cloud Audit logs, log sinks, and BigQuery.

In the upcoming lab, you'll generate logs to replicate suspicious activity involving the creation and deletion of cloud resources. You'll then analyze the logs you purposefully generated using a tool known as BigQuery.

Note: In the lab activity, you'll be using two student user accounts: username 1 and username 2. You'll use username 1 to generate the log activity and username 2 to analyze the activity in BigQuery.

Cloud Audit Logs

So far, you've learned about log tools like Cloud Logging and Logs Explorer. As a quick refresher, Cloud Logging collects logs from Google Cloud resources, which can then be accessed with Logs Explorer. You also explored using these tools to access different types of logs, like Firewall Rule logs which record the activity involving firewall rules. In the upcoming lab, you'll focus on Cloud Audit Logs which record administrative activities and accesses within your Google Cloud resources.

Cloud Audit Logs provide valuable insights into the user activity, system changes, and configuration changes within your Google Cloud resources, helping you answer investigative questions, like "Who did what?", "When?", and "Where?".

In the lab activity, you'll generate Cloud Audit logs twice. The first time, you'll purposefully generate activity that will create and remove cloud resources. The second time, you'll generate log activity that will be exported to BigQuery.

Here is an example of `gcloud` commands that create and remove cloud resources including: Cloud Storage bucket, a sample file, virtual private cloud (VCP) network, and a virtual machine (VM):

```
Unset
gcloud storage buckets create gs://$DEVSHELL_PROJECT_ID
echo "this is a sample file" > sample.txt
```

```
gcloud storage cp sample.txt gs://$DEVSHELL_PROJECT_ID
gcloud compute networks create mynetwork --subnet-mode=auto
export ZONE=$(gcloud compute project-info describe \
--format="value(commonInstanceMetadata.items[google-compute-defaul
t-zone])")
gcloud compute instances create default-us-vm \
--machine-type=e2-micro \
--zone=$ZONE --network=mynetwork
gcloud storage rm --recursive gs://$DEVSHELL_PROJECT_ID
```

Here's what each line of this command does:

- gcloud storage buckets create gs://$DEVSHELL_PROJECT_ID

This command creates a Cloud Storage bucket.

- echo "this is a sample file" > sample.txt

This command creates a file named sample.txt containing the strings this is a sample file.

- gcloud storage cp sample.txt gs://$DEVSHELL_PROJECT_ID

This command copies the sample.txt file into the Cloud Storage bucket.

- gcloud compute networks create mynetwork --subnet-mode=auto

  export ZONE=$(gcloud compute project-info describe \

  --format="value(commonInstanceMetadata.items[google-compute-default-zone])")

This command creates a VPC network named mynetwork configured as an auto mode network. ZONE sets an environment variable which contains information about the default zone for the Google Cloud project.

- gcloud compute instances create default-us-vm \

  --machine-type=e2-micro \

  --zone=$ZONE --network=mynetwork

This command creates a virtual machine (VM) named default-us-vm and sets its machine type setting, zone, and network.

- gcloud storage rm --recursive gs://$DEVSHELL_PROJECT_ID

This command deletes the Cloud Storage bucket and all of its contents.

Export logs with a sink in Google Cloud

A sink controls how Cloud Logging routes logs. With sinks, you can route some or all of your logs to a desired destination. In the lab, you'll create a sink to route logs to BigQuery and leverage its big-data analysis capabilities on your logs.

To create a log sink:

1. In the Google Cloud console, click the **Navigation** menu.
2. Click **Logging** > **Logs Explorer**.
3. Copy and paste the query provided in the lab instructions into the query builder.
4. Under the **Query editor** field, click **More actions** > **Create sink**.
5. Specify the settings of the sink, including the sink name, the sink destination to BigQuery, and the logs you want to export.
6. Click **Create Sink**.

Once the sink is created, all future logs export to BigQuery, where you can perform analysis on the Cloud Audit log data. The log export does not export existing log entries which is why you'll generate additional Cloud Activity logs a second time using another set of gcloud commands.

Note: When you configure any new sink to export logs, the current filter in the query builder is applied to what you export. If you don't specify filters in the query builder then all logs are routed to the sink's destination.

BigQuery

In the corresponding lab, you'll use gcloud commands to create and delete a Cloud Storage bucket and a VM. Deleting these cloud resources generates Admin Activity logs which are a type of Cloud Audit log that record administrative actions, like configuration changes to cloud resources. You'll then use BigQuery to analyze the Admin Activity logs.

Recall that BigQuery is a data warehouse on Google Cloud used to query data, filter large datasets, aggregate results, and perform complex operations. BigQuery uses SQL (Structured Query Language) which is a programming language used to create, interact with, and request

information from a database. In cloud cybersecurity, you can use BigQuery to perform analysis on logs and it is especially helpful when you need to work with large and complicated logs.

The following is an example of a BigQuery query you'll use in the lab:

```
Unset
SELECT
  timestamp,
  resource.labels.instance_id,
  protopayload_auditlog.authenticationInfo.principalEmail,
  protopayload_auditlog.resourceName,
  protopayload_auditlog.methodName
FROM
`auditlogs_dataset.cloudaudit_googleapis_com_activity_*`
WHERE
  PARSE_DATE('%Y%m%d', _TABLE_SUFFIX) BETWEEN
  DATE_SUB(CURRENT_DATE(), INTERVAL 7 DAY) AND
  CURRENT_DATE()
  AND resource.type = "gce_instance"
  AND operation.first IS TRUE
  AND protopayload_auditlog.methodName =
"v1.compute.instances.delete"
ORDER BY
  timestamp,
  resource.labels.instance_id
LIMIT
  1000;
```

Here's what each line of this query does:

- `SELECT`

`SELECT` indicates which columns to return. In this example, it's returning multiple columns, which include the specific information to retrieve including `timestamp` and `resource.labels.instance_id`.

- `FROM`

FROM specifies where BigQuery should look for the data. Here, it's looking within a dataset named auditlogs_dataset.cloudaudit_googleapis_com_activity_* These tables contain the Cloud Audit Logs containing the information about the actions you generated.

- `WHERE`

`WHERE` indicates the condition for a filter which returns the relevant data with the criteria that's specified.

- PARSE_DATE('%Y%m%d', _TABLE_SUFFIX) BETWEEN

  DATE_SUB(CURRENT_DATE(), INTERVAL 7 DAY) AND

  CURRENT_DATE()

In this example `WHERE` is filtering the data to only include events that occurred within the last 7 days.

- `AND resource.type = "gce_instance`

This filters further for events related to Google Compute Engine instances.

- `ORDER BY`

ORDER BY orders the results in a specific order. In this example, it's sorting the events by their timestamp and resource.labels.instance_id so that the results are ordered chronologically and grouped by their instance ID.

- LIMIT

LIMIT restricts the output to only the first 1000 results.

Key takeaways

Logs are valuable assets for cloud security analysts to review in a cloud service like Google Cloud's Cloud Audit Logs. There, you can view detailed activity records involving sensitive actions on cloud resources like the removal of Cloud Storage buckets. Tools like BigQuery can be powerful in helping you analyze logs and identify potential threats.

Resources for more information

To learn more about query BigQuery syntax, check out BigQuery Query syntax.