Google Cloud

Lab - Using your Access Token

Wherein ...

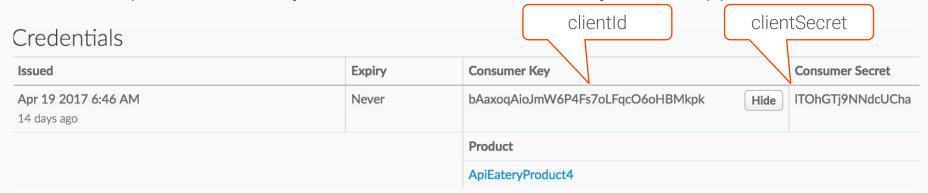
- We realize that APP Keys are still not as secure as we would like, and
- Deploy an "Oauth" proxy to generate limited time tokens for us, and
- Reconfigure our Verification to use a token, instead of an API Key

Google Cloud

Proprietary + Confidentia

Oauth proxy needs a Basic Authentication header

- Generate a base64 encoded clientId:clientSecret to receive an access_token
 - google "base64 encode"; select an online encoder
 - clip Consumer Key and Consumer Secret from your DevApp:



- encode <u>clientId:clientSecret</u>, and clip resulting encoded string
- this is "<your_b64_encoded_clientId:clientSecret>"

Google Cloud

Proprietary + Confidential

Using Access Tokens

Pass the b64encoded string as a Basic Auth Token to retrieve an access token:

Request: POST

https://<yourOrg>-test.apigee.net/oauth/client_credential/accesstoken?grant_type=client_credentials

Header:

Authorization: Basic <your_b64_encoded_clientId:clientSecret⇒

- Capture the access_token:
 - e.g.: "access_token" : "4WCAchNNtVyK8JsACl1HP7mlWW1X",

Google Cloud

Proprietary + Confidentia

Using Access Tokens

 Replace the Validate API Key policy with an Oauth Policy to validate the access_token

Google Cloud

Proprietary + Confidential

Test

• Add the access token as "Authorization: Bearer " header to your request:

Request:

http://<yourOrg>-test.apigee.net/v1/apieatery/chefs

Header:

Authorization: Bearer 4WCAchNNtVyK8JsACl1HP7mlWW1X

- Test your API call
- Once you have validated an access_token successfully, remove the Oauth Validation, and replace it with the APIKey Validation policy
 - (So we don't have to keep getting new tokens for the remaining labs...

Google Cloud

Proprietary + Confidentia

THANK YOU