Google Cloud

Lab - Threat Protection

Wherein ...

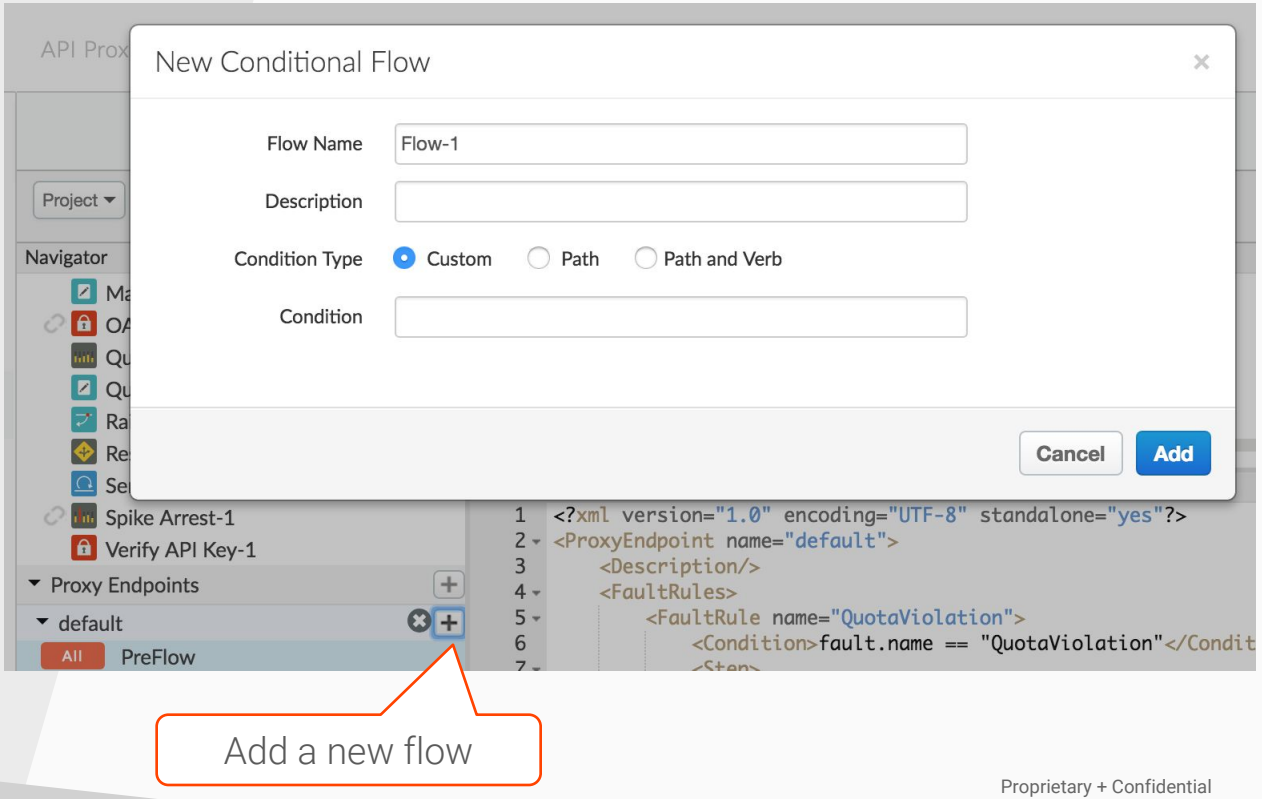
- We would like to implement checks on inbound JSON payloads to insure they conform to our known limits,

- Thereby protecting our systems from JSON payload threats

Create a POST resource or Conditional flow

Create a new POST resource in the Proxy Endpoint

- Condition Type:
  - path and verb
  - Verb: POST --
  - Path: /chefs

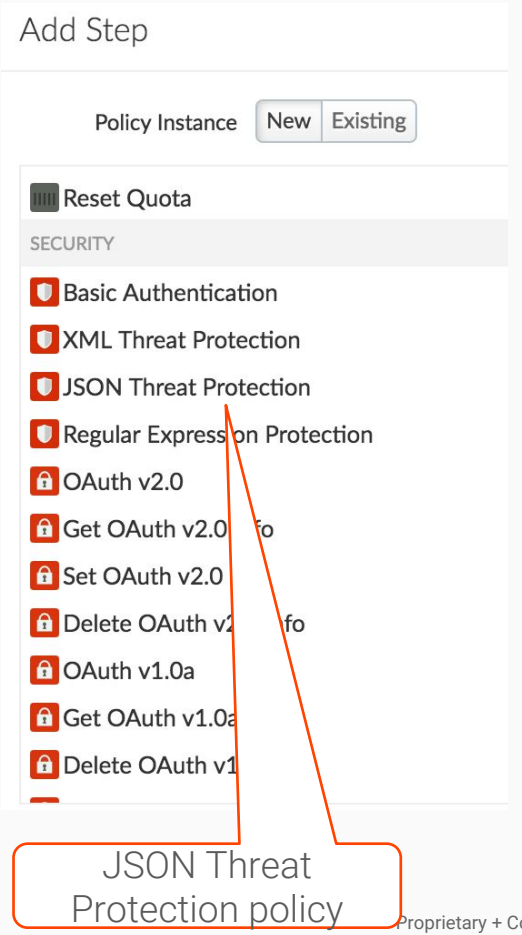


Add JSON Threat Protection

- Add the JSON Threat protection policy to the [request] POST flow that you created :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<JSONThreatProtection async="false" continueOnError="false"
enabled="true" name="JSON-Threat-Protection-1">
    <DisplayName>JSON Threat Protection-1</DisplayName>
    <Properties/>
    <ArrayElementCount>0</ArrayElementCount>
    <ContainerDepth>1</ContainerDepth>
    <ObjectEntryCount>4</ObjectEntryCount>
    <ObjectEntryNameLength>50</ObjectEntryNameLength>
    <Source>request</Source>
    <StringValueLength>250</StringValueLength>
</JSONThreatProtection>
```

- Change the default configuration options in the policy to match the above values



Test

- Make sure to set http header to **Content-Type: application/json**
- A Chef object is shown below and we need to make sure all requests adhere to this format.

NOTE: Change the name attribute value to something you can identify with later.

{ "name":”SteveJohnson", "education":"Street", "description":"TBD", "tagline":"I love me some hamburgers testing my api” }

- Test the endpoint with the following additional payloads:

{ "name":"SteveJohnson", "education":"Street", "description":"TBD", "tagline":"I love me some hamburgers testing my api", "hack":"Injecting bad burger!" }

{ "badParent" : { "name":"SteveJohnson", "education":"Street", "description":"TBD", "tagline":"I love me some hamburgers testing my api" } }

THANK YOU