

experiment Lab schedule 1 hour 30 minutes universal_currency_alt No cost

show_chart Introductory

Overview

In this lab, you use the `RegularExpressionProtection` policy to detect malicious requests.

Objectives

In this lab, you learn how to protect your API proxy against malicious requests using regular expressions.

Setup

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, **1:15:00**), and make sure you can finish within that time.
There is no pause feature. You can restart if needed, but you have to start at the beginning.
3. When ready, click **Start lab**.

4. Note your lab credentials (**Username** and **Password**). You will use them to sign in to the Google Cloud Console.
5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.
If you use other credentials, you'll receive errors or **incur charges**.
7. Accept the terms and skip the recovery resource page.

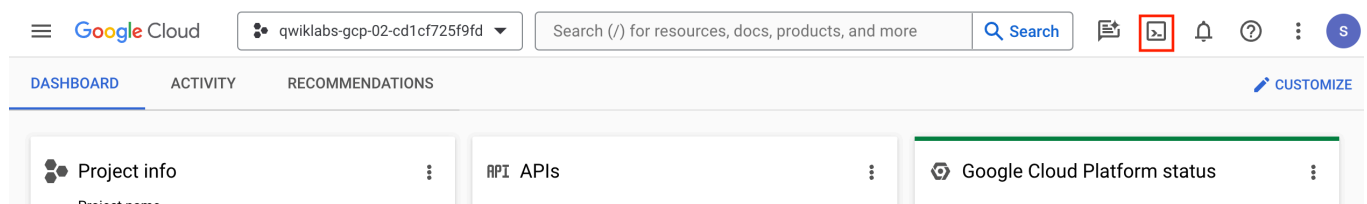
Note: Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

Activate Google Cloud Shell

Google Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud.

Google Cloud Shell provides command-line access to your Google Cloud resources.

1. In Cloud console, on the top right toolbar, click the Open Cloud Shell button.



2. Click **Continue**.

It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:

```
...abs-gcp-44776a13dea667a6) x + ▾
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6) $
```

gcloud is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

- You can list the active account name with this command:

```
gcloud auth list
```

Output:

```
Credentialed accounts:
- @.com (active)
```

Example output:

```
Credentialed accounts:
- google1623327_student@qwiklabs.net
```

- You can list the project ID with this command:

```
gcloud config list project
```

Output:

```
[core]
project =
```

Example output:

[core]

```
project = qwiklabs-gcp-44776a13dea667a6
```

Note: Full documentation of **gcloud** is available in the gcloud CLI overview guide .

Task 1. Create a new proxy

In this task, you create a new API proxy.

1. In the Google Cloud console, on the **Navigation menu** (≡), select **Integration Services > Apigee > Proxy Development > API proxies**.
2. To start the proxy wizard, click **+Create**.
3. Leave **Proxy template** unchanged.
4. Specify the following settings:

| Property | Value |
|-----------------------|-------------------------------------|
| Proxy Name | lab5a-v1 |
| Base path | /lab5a/v1 |
| Target (Existing API) | https://httpbin.org/anything |

The **httpbin.org/anything** API returns detailed information about the API request it was sent.

Note: Confirm that you are using **"/lab5a/v1"** for the base path, and not **"/lab5a-v1"**.

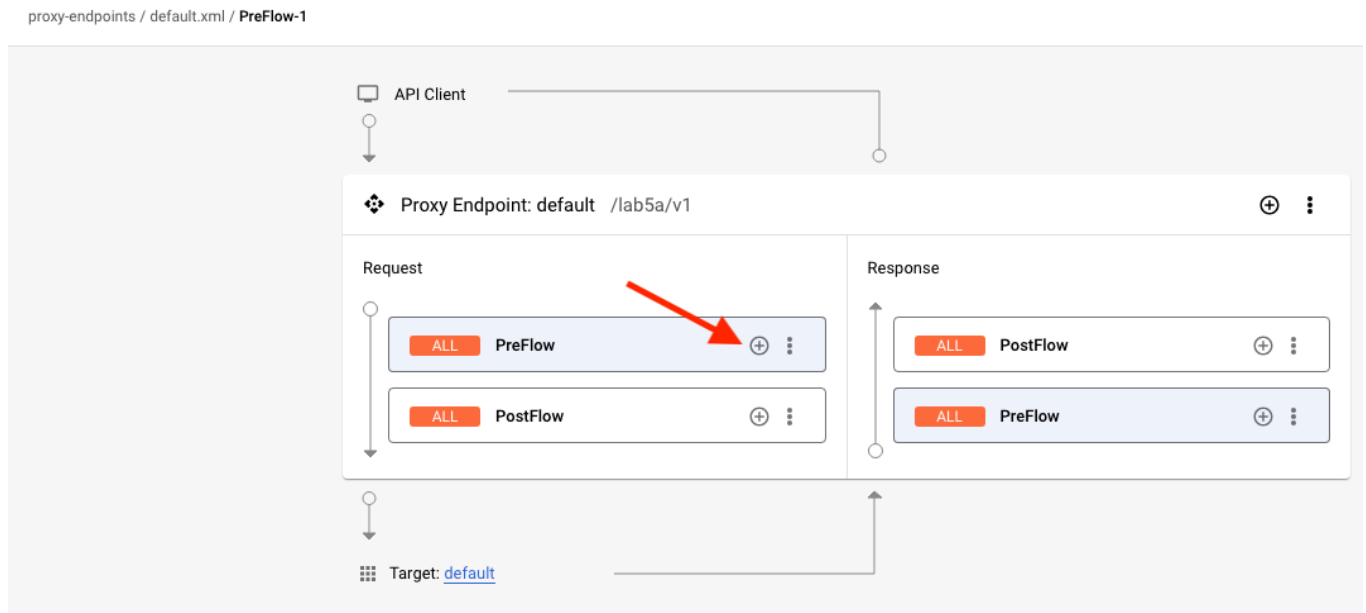
5. Click **Create**.

6. Click the **Develop** tab.

Task 2. Add a RegularExpressionProtection policy

In this task, you add a RegularExpressionProtection policy to protect against malicious requests.

1. Click **Proxy endpoints > default > PreFlow**.
2. On the **Request PreFlow**, click **Add Policy Step (+)**.



3. In the **Add policy step** pane, select **Create new policy**, and then select **Security > Regular Expression Protection**.
4. Specify the following values:

| Property | Value |
|--------------|-----------------------------|
| Name | RegexTP-SQLInjection |
| Display name | RegexTP-SQLInjection |

5. Click **Add**.

6. Click **Policies > RegexTP-SQLInjection**.

7. Replace the policy's default configuration with:

```
<RegularExpressionProtection continueOnError="false"
enabled="true" name="RegexTP-SQLInjection">
  <IgnoreUnresolvedVariables>false</IgnoreUnresolvedVariables>
  <QueryParam name="test">
    <Pattern>[\s]*(?i)((delete)|(exec)|(drop\s*table)|(insert)|
(shutdown)|(update)|(\bor\b))</Pattern>
  </QueryParam>
  <Source>request</Source>
</RegularExpressionProtection>
```

content_co

A RegularExpressionProtection policy raises a fault if any of the regular expressions in the policy match the data in the configured location.

The **Source** is set to **request**. A RegularExpressionProtection policy can validate many parts of the request. In this case, the policy is configured to check the query parameter named **test** using a regular expression that contains several dangerous SQL patterns.

Note: The pattern configured in the proxy is an example regular expression intended for use in this lab. It is not intended to recommend a pattern that would protect your proxies in a production environment. You should consult your security team to determine the regular expressions necessary to protect your APIs for your specific scenarios.

8. Click **Save**.

9. Click **Deploy**.

10. To specify that you want the new revision deployed to the eval environment, select **eval** as the **Environment**, and then click **Deploy**.

11. Click **Confirm**.

Check deployment status

A proxy that is deployed and ready to take traffic will show a green status.

Deployments

| Status | Revision | Environment | |
|--------|----------|-------------|----------|
| ✓ | 1 | eval | UNDEPLOY |

✓ eval

✕

Status: Deployed

Deployed on: 11/27/2023, 6:19:20 PM

| Revision | Description |
|----------|---------------|
| 1 | My retail API |

When a proxy is marked as deployed but the runtime is not yet available and the environment is not yet attached, you may see a red warning sign. Hold the pointer over the **Status** icon to see the current status.

Google Cloud

qwiklabs-gcp-03-91df34bcb84e

Search (/) for resources, docs, products, and more

Apigee

Overview

Proxy development

API proxies

Shared flows

Integrations

Offline debug

API monitor

Distribution

API product

Portals

Developers

retail-v1

OVERVIEW

DEVELOP

DEBUG

Proxy summary

Deployments

1 Disruption

| Status | Revision | Environment | |
|--------|----------|-------------|----------|
| ! | 1 | eval | UNDEPLOY |

! eval

✕

Status: no instances are reporting status for this environment

Deployed on: 11/27/2023, 6:19:20 PM

| | Description | Last modified |
|---|---------------|----------------|
| 1 | My retail API | November 27, : |

If the proxy is deployed and shows as green, your proxy is ready for API traffic. If your proxy is not deployed because there are no runtime pods, you can check the status of provisioning.

Check provisioning dashboard

1. In the Google Cloud Console, navigate to **Compute Engine > VM instances**.
2. To open the Lab Startup Tasks dashboard, click on the **External IP** for the **lab-startup** VM.

Compute Engine

Virtual machines

VM instances

Instance templates

Sole-tenant nodes

Machine images

TPUs

Committed use discounts

VM instances

CREATE INSTANCE

IMPORT VM

REFRESH

INSTANCES

OBSERVABILITY

INSTANCE SCHEDULES

Filter

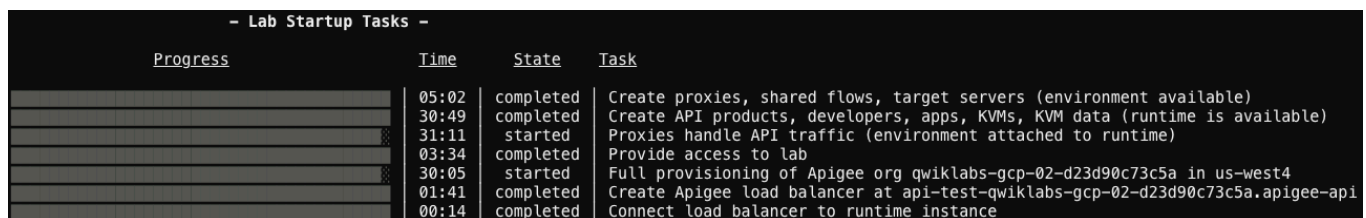
Enter property name or value

| <input type="checkbox"/> | Status | Name ↑ | Zone | Recommendations | In use by | Internal IP | External IP | Connect |
|--------------------------|--------|-----------------------------------|---------------|-----------------|------------------------------------|-------------------|--------------------------------------|---------|
| <input type="checkbox"/> | ✓ | apigee-proxy-07m8 | us-central1-f | | apigee-proxy-group | 10.128.0.3 (nic0) | 34.27.73.218 (nic0) | SSH ▾ ⋮ |
| <input type="checkbox"/> | ✓ | apigee-proxy-dg85 | us-central1-f | | apigee-proxy-group | 10.128.0.4 (nic0) | 34.132.15.243 (nic0) | SSH ▾ ⋮ |
| <input type="checkbox"/> | ✓ | apigee-proxy-gdxx | us-central1-f | | apigee-proxy-group | 10.128.0.5 (nic0) | 35.188.25.205 (nic0) | SSH ▾ ⋮ |
| <input type="checkbox"/> | ✓ | lab-startup | us-central1-f | | | 10.128.0.2 (nic0) | 34.28.102.86 (nic0) | SSH ▾ ⋮ |

3. If you see a redirect notice page, click the link to the external IP address.

A new browser window will open. Lab startup tasks are shown with their progress.

- *Create proxies, shared flows, target servers* should be complete when you first enter the lab, allowing you to use the Apigee console for tasks like proxy editing.
- *Create API products, developers, apps, KVMs, KVM data* indicates when the runtime is available and those assets may be saved.
- *Proxies handle API traffic* indicates when the eval environment has been attached to the runtime and the deployed proxies can take runtime traffic.



| - Lab Startup Tasks - | | | |
|-----------------------|-------|-----------|---|
| Progress | Time | State | Task |
| <div></div> | 05:02 | completed | Create proxies, shared flows, target servers (environment available) |
| <div></div> | 30:49 | completed | Create API products, developers, apps, KVMs, KVM data (runtime is available) |
| <div></div> | 31:11 | started | Proxies handle API traffic (environment attached to runtime) |
| <div></div> | 03:34 | completed | Provide access to lab |
| <div></div> | 30:05 | started | Full provisioning of Apigee org qwiklabs-gcp-02-d23d90c73c5a in us-west4 |
| <div></div> | 01:41 | completed | Create Apigee load balancer at api-test-qwiklabs-gcp-02-d23d90c73c5a.apigee-api |
| <div></div> | 00:14 | completed | Connect load balancer to runtime instance |

In this case, you need to wait for *Proxies handle API traffic* to complete.

While you are waiting

While you wait for the lab to start up, learn more about the policy and regular expressions:

- Documentation for the policy — [RegularExpressionProtection policy reference](#)
- Entry for regular expressions, containing good examples of the syntax — [Wikipedia: Regular Expressions](#)

Task 3. Test the API proxy

In this task, you use curl to test your proxy.

1. In Cloud Shell, send the following curl command:

```
curl -X GET "https://api-  
test-${GOOGLE_CLOUD_PROJECT}.apiservices.dev/lab5a/v1" | json_pp
```

content_c

The curl command responds successfully with the response from httpbin.org. There was no query parameter named *test*, so there was no invalid pattern detected.

2. Send the following curl command:

```
curl -X GET "https://api-  
test-${GOOGLE_CLOUD_PROJECT}.apiservices.dev/lab5a/v1?test=ok" |  
json_pp
```

content_c

This command sends the test query parameter with the value *ok*. This does not match the regular expression in the policy, so the policy does not raise a fault, and the curl command responds successfully with the response from httpbin.org.

3. Send the following curl command:

```
curl -X GET "https://api-  
test-${GOOGLE_CLOUD_PROJECT}.apiservices.dev/lab5a/v1?"
```

content_c

```
test=delete" | json_pp
```

This command sends the test query parameter with the value *delete*. This matches the regular expression in the policy, so this time the policy raises a fault, and `httpbin.org` is never called.

Note: The default status code for the error, 500 Internal Server Error, should typically be changed. This error is occurring because an illegal request was sent by the client, so 400 Bad Request would be a better status code to return.

It is typically appropriate to rewrite the error message as well, because the regular expression that matches the request is shown in the error response (as shown above), and you typically do not want a malicious actor to know the regular expressions that are protecting your proxy.

Congratulations!

In this lab, you used the `RegularExpressionProtection` policy to detect a dangerous input and reject the request.

End your lab