Using Auth0 to authenticate users

cloud.google.com/api-gateway/docs/authenticating-users-auth0

This page describes how to support user authentication in API Gateway.

To authenticate a user, a client application must send a JSON Web Token (JWT) in the authorization header of the HTTP request to your backend API. API Gateway validates the token on behalf of your API, so you don't have to add any code in your API to process the authentication. However, you do need to configure the API config for your gateway to support your chosen authentication methods.

API Gateway validates a JWT in a performant way by using the JWT issuer's JSON Web Key Set (JWKS). The location of the JWKS is specified in the `x-google-jwks_uri` field of the gateway's API config. API Gateway caches the JWKS for five minutes and refreshes it every five minutes.

Before you begin

Add authentication code to your client application, following the Auth0 documentation.

When your client application sends an HTTP request, the authorization header in the request must contain the following JWT claims:

- `iss` (issuer)
- `sub` (subject)
- `aud` (audience)
- `iat` (issued at)
- `exp` (expiration time)

Configuring API Gateway to support client authentication

You must have a security requirement object and a security definitions object in your API config for API Gateway to validate the claims in the signed JWT.

To support Auth0 authentication:

1. Add the following to the security definition in your API config:

```
securityDefinitions:
   auth0_jwk:
     authorizationUrl: ""
     flow: "implicit"
     type: "oauth2"
     # Replace YOUR-ACCOUNT-NAME with your Auth0 account name.
     x-google-issuer: "https://YOUR-ACCOUNT-NAME.auth0.com/"
     x-google-jwks_uri: "https://YOUR-ACCOUNT-NAME.auth0.com/.well-
known/jwks.json"
     # Optional. Replace YOUR-CLIENT-ID with your client ID
     x-google-audiences: "YOUR-CLIENT-ID"
```

2. Add a security section at either the API level to apply to the entire API, or at the method level to apply to a specific method.

```
security:
   - auth0_jwk: []
```

You can define multiple security definitions in the API config, but each definition must have a different issuer. If you use security sections at both the API level and at the method level, the method-level settings override the API-level settings.

The `x-google-audiences` field isn't required. API Gateway accepts all JWTs with the backend service name in the form of `https://SERVICE_NAME` in the `aud` claim. To allow additional client IDs to access the backend service, you can specify the allowed client IDs in the `x-google-audiences` field by using comma-separated values. API Gateway then accepts the JWTs with any of the specified client IDs in the `aud` claim.

Making an authenticated call to an API Gateway API

When you send a request using an authentication token, we recommend that you put the token in the `Authorization:Bearer` header. For example:

```
curl --request POST \
  --header "Authorization: Bearer ${TOKEN}" \
  "${GATEWAY_URL}/echo"
```

Here, `GATEWAY_URL` and `TOKEN` are environment variables containing your deployed gateway URL and authentication token, respectively. See Making an authenticated request to an API Gateway API for sample code that sends a request using the `Authorization:Bearer` header.

If you cannot use the header when sending the request, you can put the authentication token in a query parameter called `access_token`. For example:

```
curl "${GATEWAY_URL}/echo?access_token=${TOKEN}"
```

Receiving authenticated results in your API

API Gateway usually forwards all headers it receives. However, it overrides the original `Authorization` header when the backend address is specified by `x-google-backend` in the API config.

API Gateway will send the authentication result in the `X-Apigateway-Api-Userinfo` to the backend API. It is recommended to use this header instead of the original `Authorization` header. This header is `base64url` encoded and contains the JWT payload.

What's next

Authentication between services

Rate and review