**Istio Custom Authorization**

**Authorization architecture**

The authorization policy enforces access control to the inbound traffic in the server side Envoy proxy.

Each Envoy proxy runs an authorization engine that authorizes requests at runtime. When a request comes to the proxy, the authorization engine evaluates the request context against the current authorization policies, and returns the authorization result, either ALLOW or DENY.

Operators specify Istio authorization policies using .yaml files.

You don’t need to explicitly enable Istio’s authorization features; they are available after installation. To enforce access control to your workloads, you apply an authorization policy.

For workloads without authorization policies applied, Istio allows all requests.

### Authorization policies

To configure an authorization policy, you create an [AuthorizationPolicy custom resource](https://istio.io/latest/docs/reference/config/security/authorization-policy/). An authorization policy includes a selector, an action, and a list of rules:

* The selector field specifies the target of the policy
* The action field specifies whether to allow or deny the request
* The rules specify when to trigger the action
  + The from field in the rules specifies the sources of the request
  + The to field in the rules specifies the operations of the request
  + The when field specifies the conditions needed to apply the rule

The following example shows an authorization policy that allows two sources, the cluster.local/ns/default/sa/sleep service account and the dev namespace, to access the workloads with the app: httpbin and version: v1 labels in the foo namespace when requests sent have a valid JWT token.

apiVersion: security.istio.io/v1beta1

kind: AuthorizationPolicy

metadata:

name: httpbin

namespace: foo

spec:

selector:

matchLabels:

app: httpbin

version: v1

action: ALLOW

rules:

- from:

- source:

principals: ["cluster.local/ns/default/sa/sleep"]

- source:

namespaces: ["dev"]

to:

- operation:

methods: ["GET"]

when:

- key: request.auth.claims[iss]

values: ["https://accounts.google.com"]

The following example shows an authorization policy that denies requests if the source is not the foo namespace:

apiVersion: security.istio.io/v1beta1

kind: AuthorizationPolicy

metadata:

name: httpbin-deny

namespace: foo

spec:

selector:

matchLabels:

app: httpbin

version: v1

action: DENY

rules:

- from:

- source:

notNamespaces: ["foo"]

The deny policy takes precedence over the allow policy. Requests matching allow policies can be denied if they match a deny policy. Istio evaluates deny policies first to ensure that an allow policy can’t bypass a deny policy.