Test preliminari per le Istio Policy

Ai fini di testare la corretta applicazione delle Istio Policy per I namespace di FREE, occorre provare che la comunicazione prima dell’applicazione delle suddette policy sia consentita all’interno del cluster fra tutti I servizi di tutti I namespace. A tal fine si predispongono i seguenti test:

# Comunicazione pre-installazione Istio Policy

Prima di applicare le policy Istio qualsiasi tipo di comunicazione inter-pod o pod-service dev’essere consentita come da prassi Kubernetes. Per cui occorre provare che dato un pod di un qualsiasi namespace (Frontend, MGW, BE) è funzionante la comunicazione diretta verso un qualsiasi altro pod dello stesso namespace o altro namespace.

Le comunicazioni, che devono avvenire tra un pod e il service del pod destinazione, possono essere testate con un semplice comando curl. Non è necessaria coerenza applicativa tra le chiamate. Basta una qualsiasi chiamata verso il servizio esposto dal destinatario anche se porta ad un errore applicativo, con il solo scopo di verificare la corretta apertura del canale di comunicazione.

Procedura:

1. Collegarsi in ssh al pod sorgente
2. Dal pod eseguire

curl http://<servicename>.<namespace>.svc.cluster.local:<port>

es. curl <http://b-be-1.b-be.svc.cluster.local:8080>

es. Curl HTTPS <https://b-be-1.b-be.svc.cluster.local>:443

Eseguire quindi I seguenti test:

* BFF -> BFF (stesso namespace) - Esito atteso: comunicazione consentita
* BE -> BE (stesso namespace) - Esito atteso: comunicazione consentita
* BFF->BE (stesso RGP) - Esito atteso: comunicazione consentita
* BFF->BE (RGP diversi) - Esito atteso: comunicazione consentita
* BE->BE (RGP diversi) - Esito atteso: comunicazione consentita

Inoltre dev’essere funzionante il normale flusso BFF->MGW->BE, può essere verificato tramite test puntuale o normale test applicativo. Esito atteso: comunicazione consentita come da normale AS-IS.

# Comunicazione post-installazione Istio Policy

A valle dell’installazione delle istio policy, dell’abilitazione dell’iniezione per I namespace oggetto del test, nonché della verifica che I container di envoy sono stati correttamente iniettati.

Ri-Eseguire quindi I test:

* BFF -> BFF (stesso namespace) - Esito atteso: comunicazione consentita
* BE -> BE (stesso namespace) - Esito atteso: comunicazione consentita
* BFF->BE (stesso RGP) - Esito atteso: **comunicazione negata**
* BFF->BE (RGP diversi) - Esito atteso: **comunicazione negata**
* BE->BE (RGP diversi) - Esito atteso: **comunicazione negata**

NB: NEL CASO DI COMUNICAZIONE NEGATA A CAUSA DELL’APPLICAZIONE DI UNA ISTIO POLICY SI DOVRA’ OTTENERE IL SEGUENTE MESSAGGIO DI ERRORE:

**RBAC: access denied**

Le stesse comunicazioni di cui sopra dovranno però funzionare utilizzando il flusso previsto che passa dai microgateway:

* BFF->MGW->BE (stesso RGP) - Esito atteso: comunicazione consentita
* BFF->MGW->BE (RGP diversi) - Esito atteso: comunicazione consentita
* BE->MGW->BE (RGP diversi) - Esito atteso: comunicazione consentita

Per eseguire I test ove previsto il flusso applicativo basterà il test applicativo. Ove non previsto o non disponibile, si lancerà una curl diretta.

Es. test di comunicazione tra il BFF dell’RGPa e il BE dell’RGPb:

* ssh su pod BFF dell’RGPa
* curl service mgw dell’RGPa:
  + curl <http://a-mgw-1.a-mgw.svc.cluster.local>:MGW-PORT
* ssh su pod mgw dell’RGPa
* curl su service BE dell’RGPb:
  + curl <http://b-be-1.b-be.svc.cluster.local>:MGW-PORT

L’esito atteso sarà di **comunicazione consentita**.