**Verifica certificato CA in mTLS = Permissive**  
  
E' possibile scaricare dall'istio-proxy l'intera configurazione envoy con il seguente comando:  
```  
oc -n <<namespace>> exec -it pods/<<pod-name>> -c istio-proxy -- curl localhost:15000/config\_dump >istio\_proxy\_config\_dump.json  
```  
  
A questo punto è possibile individuare la rootCA e la trusted chain all'interno del file di configurazione (search for `ROOTCA` e `certificate\_chain`)  
  
E' poi necessario decodificare l'attributo inline\_bytes (`base64 --decode`) e confrontarli con `root-cert-pem` e `cert-chain.pem` del `cacerts` secret in `istio-system`  
  
Il seguente comando alternativo decodifica direttamente la ROOTCA (necessario `jq`)  
```  
cat istio\_proxy\_config\_dump.json | jq -r '.configs[] | select(.["@type"] == "[type.googleapis.com/envoy.admin.v3.SecretsConfigDump](https://urldefense.com/v3/__http:/type.googleapis.com/envoy.admin.v3.SecretsConfigDump__;!!OrxsNty6D4my!_k0u0c8KT57QmXWyVulVUd3CBKEgyGLOPqdNaK123svS0wmBeY3tdVS4qDEWmyDRd51BGd70dco19bvC21E8HA4$)") | .dynamic\_active\_secrets[] | select(.name == "ROOTCA") | .secret.validation\_context.trusted\_ca.inline\_bytes' | base64 --decode  
```  
  
Disabilitare mTLS per uno specifico servizio  
  
**Destination Rule**   
Fare riferimento alla documentazione a piè di pagina  
```  
apiVersion: [networking.istio.io/v1beta1](https://urldefense.com/v3/__http:/networking.istio.io/v1beta1__;!!OrxsNty6D4my!_k0u0c8KT57QmXWyVulVUd3CBKEgyGLOPqdNaK123svS0wmBeY3tdVS4qDEWmyDRd51BGd70dco19bvCBhhf_uM$)  
kind: DestinationRule   
metadata:   
  name: no-mtls-service1  
spec:   
  host: service1.namespace.svc.cluster.local   
  trafficPolicy:   
    tls:   
      mode: DISABLE  
```

Definizione Servizi esterni   
  
**Service Entry**  
Fare riferimento alla documentazione a piè di pagina  
```  
apiVersion: [networking.istio.io/v1beta1](https://urldefense.com/v3/__http:/networking.istio.io/v1beta1__;!!OrxsNty6D4my!_k0u0c8KT57QmXWyVulVUd3CBKEgyGLOPqdNaK123svS0wmBeY3tdVS4qDEWmyDRd51BGd70dco19bvCBhhf_uM$)  
kind: ServiceEntry  
metadata:   
  name: external-service  
spec:   
  hosts:   
    - external-service.domain.internal   
  ports:  
    - number: 80  
      name: http  
      protocol: HTTP  
  resolution: DNS  
  location: MESH\_EXTERNAL  
```  
  
**Destination Rule**   
  
```  
apiVersion: [networking.istio.io/v1beta1](https://urldefense.com/v3/__http:/networking.istio.io/v1beta1__;!!OrxsNty6D4my!_k0u0c8KT57QmXWyVulVUd3CBKEgyGLOPqdNaK123svS0wmBeY3tdVS4qDEWmyDRd51BGd70dco19bvCBhhf_uM$)  
kind: DestinationRule   
metadata:   
  name: no-mtls-external-service  
spec:   
  host: external-service.domain.internal  
  trafficPolicy:  
    tls:   
      mode: DISABLE  
```  
  
*Nota*: da tenere presente che le risorse `ServiceEntry` e `DestinationRule` sono visibili da tutta la mesh. Per limitare la visibilità è necessario specificare esplicitamente l'elemento `exportTo`(vedi documentazione a piè di pagina)  
  
Documentazione (Istio)

* [https://istio.io/latest/docs/reference/config/networking/destination-rule/](https://urldefense.com/v3/__https:/istio.io/latest/docs/reference/config/networking/destination-rule/__;!!OrxsNty6D4my!_k0u0c8KT57QmXWyVulVUd3CBKEgyGLOPqdNaK123svS0wmBeY3tdVS4qDEWmyDRd51BGd70dco19bvC0nnoXNE$)
* [https://istio.io/latest/docs/reference/config/networking/service-entry/](https://urldefense.com/v3/__https:/istio.io/latest/docs/reference/config/networking/service-entry/__;!!OrxsNty6D4my!_k0u0c8KT57QmXWyVulVUd3CBKEgyGLOPqdNaK123svS0wmBeY3tdVS4qDEWmyDRd51BGd70dco19bvCqCFEX00$)
* [Control configuration sharing across namespaces](https://urldefense.com/v3/__https:/istio.io/latest/docs/ops/best-practices/traffic-management/*cross-namespace-configuration__;Iw!!OrxsNty6D4my!_k0u0c8KT57QmXWyVulVUd3CBKEgyGLOPqdNaK123svS0wmBeY3tdVS4qDEWmyDRd51BGd70dco19bvCYZUpSZ0$)