

# Zero Trust platform with Istio

Security BSides Krakow / 24 Sep 2022  
Marek Szlagor





# \$ whoami

- > 10 years of experience
- worked with embedded and cloud (read: shocked by carelessly rewiring devices and broke production by typo in yaml)
- last 2 years working for Remitly



---

# What we will discuss

1. What is zero trust? 
2. What is Istio? 
3. How to make it work together? 
4. Demo 

---

# Zero Trust [networking]?

1. VPN / castle-and-moat
2. Trust nothing, require every request to be authorized
3. Implement in every system? Even in this legacy php one that nobody wants to touch?
4. oauth-proxy?

---

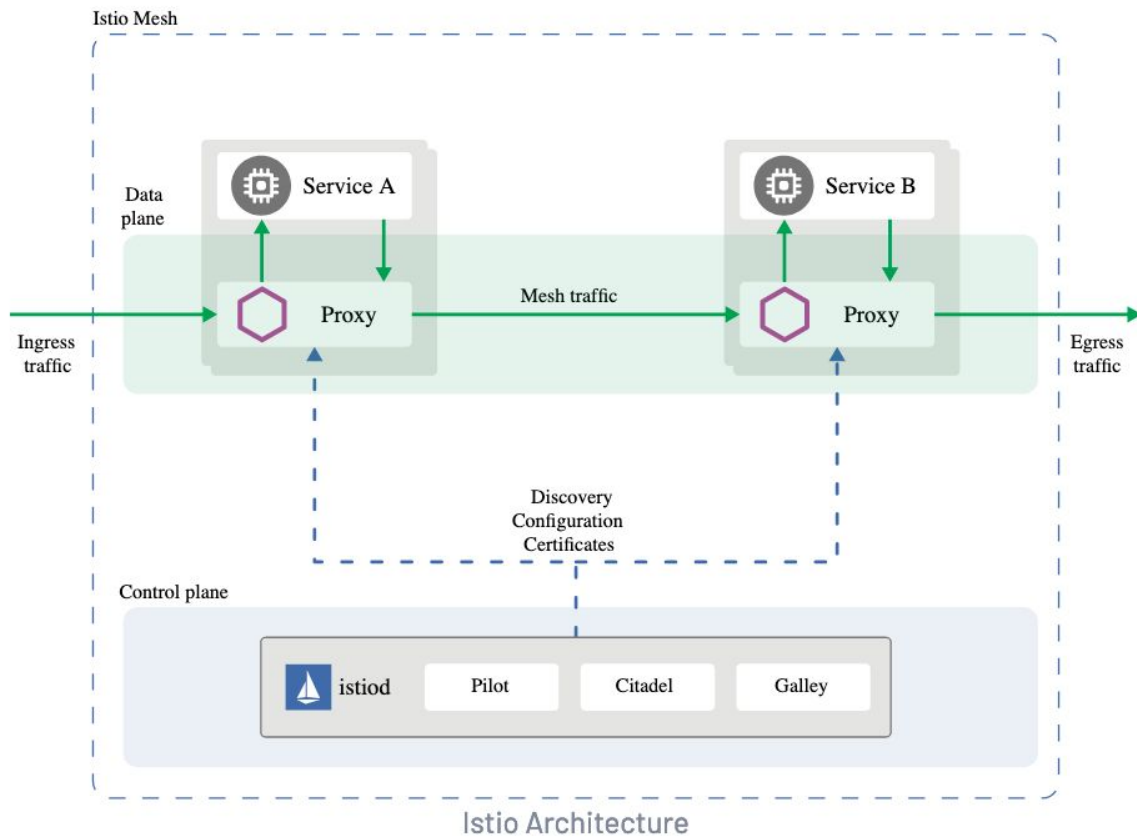
## Assumption 1

OIDC compatible identity provider in your organization

---

# Istio?

- Service mesh
- Envoy - proxy with 'hot reloads' using admin API, L7 protocols support
- Observability, traffic management, security



---

## Assumption 2

workload running on k8s cluster with Istio



---

# RequestAuthentication

Configure sidecar to check if JWT provided is:

- issued by configured issuer
- not expired
- signed by trusted key

And refuse with 401 otherwise

```
apiVersion: security.istio.io/v1beta1
kind: RequestAuthentication
metadata:
  name: nginx-requestauthentication
spec:
  selector:
    matchLabels:
      app: nginx
  jwtRules:
    - issuer: my-oidc.com
      jwksUri: https://my-oidc.com/oauth2/v1/keys
      forwardOriginalToken: true
      fromHeaders:
        - name: jwt-header-name
```

---

# AuthorizationPolicy

Configure sidecar to check if:

- request was verified by RA
- has proper header[s], proper claims
- coming from allowed pod / ingress / ip
- sent to allowed path

And refuse with 403 otherwise

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: nginx-authorizationpolicy
spec:
  selector:
    matchLabels:
      app: nginx
  rules:
    - from:
        - source:
            requestPrincipals: ["my-oidc.com/*"]
      when:
        - key: request.auth.claims[groups]
          values: ["admins", "testers"]
```

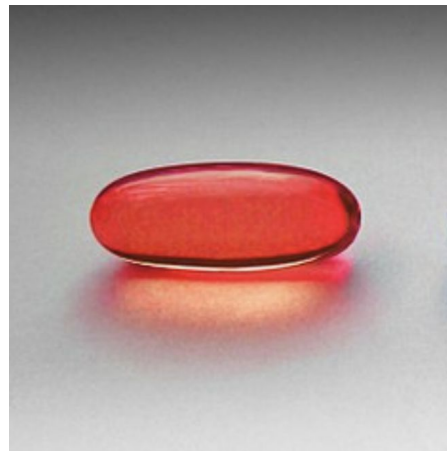


---

# EnvoyFilter

“There is no such thing as containers in kernel”

“There is no such thing as RA and AP in Istio”



---

# Envoy Filter - example Lua script

*# get jwt claims form request*

*claims = request\_handle:streamInfo():dynamicMetadata():get("istio\_authn")["request.auth.claims"]*

*# add header with user email*

*email = claims["email"]*

*request\_handle:headers():replace("x-user-email", email)*

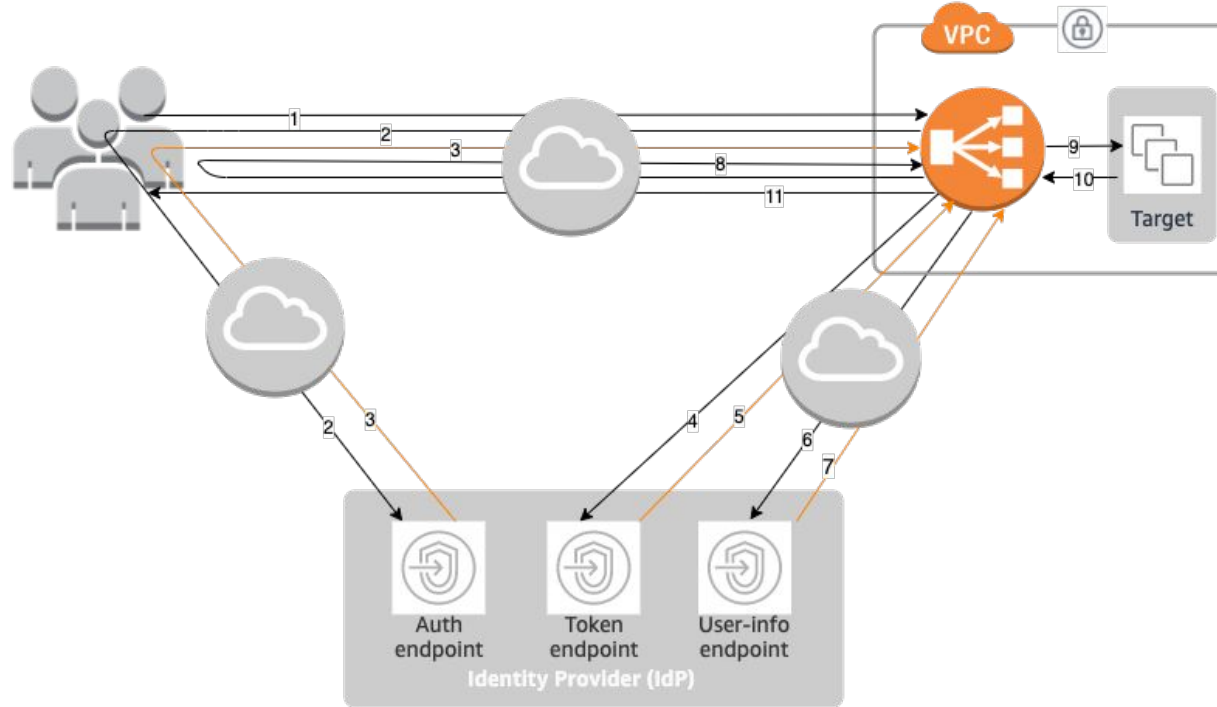
*# log*

*path = request\_handle:headers():get(":path")*

*request\_handle:logWarn({'"user": "'..email..'',"path": "'..path..''})*



# Load Balancer with OIDC



---

# Demo!

*<https://github.com/mszlgr/istio-oidc>*

---

## Let's summarize

- Claims are verified as close to application as possible
- No changes in app's code required
- Unified format of access & audit logs
- Access policy that can be shared between multiple apps