A Beginner's Guide to following Istio's security best practices

Jacob Delgado / Aspen Mesh

IstioCon

#IstioCon

Security best practices

Many users, newcomers and experienced, using Istio are using many of the defaults installed with Helm and istioctl.

- Adding security settings to a system after installation can be tedious and difficult for operators and developers alike
- Some of these settings are default because of
  - Legacy reasons (e.g. possible migration issues)
  - Ease of onboarding users
  - No one-size fits all security posture
- Many of the settings come from https://istio.io/latest/docs/ops/best-practices/security/
- There are others that are *opinionated* (and perhaps controversial!) that will be marked with *

This is not a comprehensive guide

- Istio and Kubernetes are complex pieces of software
- Prefer being explicit over relying on default, sometimes “auto” capabilities
- IT security practices vary from company to company
  - Compliance (e.g. PCI or GDPR)
  - Monitoring
  - Audits
- Do not adopt these settings without testing as changes may result in outages

mTLS should be the default traffic pattern in your service mesh*

Problem: It is *possible* (although unlikely) to serve over plaintext.

By default, clients with sidecars, are configured to use **auto-MTLS** and servers with sidecars are set to be set in **PERMISSIVE** mode.

- Permissive configures sidecars to serve over plaintext and mTLS

```
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
  namespace: istio-system
spec:
  mtls:
    mode: STRICT
—
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: default
  namespace: istio-system
spec:
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
```

Not all traffic is captured by Envoy for inbound and outbound connections

Do not depend on Istio **alone** to secure traffic internally and externally of your service mesh.

NOTE: Previously it was stated **For IPv4 clusters, IPv6 traffic is not intercepted**. This was wrong. Traffic is not allowed.

- Only TCP is captured
  - Limited support for http3 (UDP)
  - UDP and ICMP are not captured
- Some ports are not captured
  - 22 (SSH)
  - Ports specified by annotations
  - Ports used by the sidecar
- For dual-stack clusters, IPv4 traffic is intercepted, but not IPv6* (*dual-stack is NOT supported*)

Control traffic to and from your sidecars

- Use a layered approach to shaping and controlling traffic within your environment
  - Cloud/on-prem configuration
- Gateway pods should run on nodes dedicated for gateway traffic
- Use a network plugin that supports Kubernetes NetworkPolicy objects (e.g. Calico)
  - Limit ingress and egress traffic where possible

Do not depend on Istio **alone** to secure traffic internally and externally of your service mesh.

Control traffic from your service mesh and Kubernetes cluster

- Any service with a sidecar proxy is able to communicate with an external website
- By changing this to `REGISTRY_ONLY` some services may break if they are communicating with external services
  - `ServiceEntry`s must be created for each site your workloads can reach to externally
- Redirect all outbound traffic through your egress gateway if possible
  - Easier to monitor traffic

```
meshConfig:
  outboundTrafficPolicy:
    mode: REGISTRY_ONLY
```

This is useful for helping manage traffic, but should not be thought of as a firewall-like mechanism. Configure a `NetworkPolicy` and manage inbound/outbound rules for your VPC!

Disable automatic settings*

- Disable auto mtls
  - But set PeerAuthentication and DestinationRules in istio-system namespace (see earlier slides)
- Disable protocol sniffing
  - However, Istio will require Service port names to be properly prefixed to enable various functionality

```
meshConfig:
  enableAutoMtls: false
```

```
pilot:
  enableProtocolSniffingForOutbound: false
  enableProtocolSniffingForInbound: false
```

Harden your environment

- Use Istio CNI plugin
  - Reduces the privileges necessary for Istio to intercept traffic to and from your sidecar
- Use distroless images
  - Unfortunately, this can make advanced troubleshooting difficult as various tools aren’t available
  - Very little attack surface, security scanners are less noisy
- Keep up to date
  - https://istio.io/latest/docs/releases/supported-releases/

See https://istio.io/latest/docs/setup/additional-setup/cni/ for reasons why CNI uses fewer privileges to work

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  components:
    cni:
      enabled: true


global:
  tag: 1.13.2-distroless
```

RequestAuthentication and AuthorizationPolicy

- Too complicated to cover in a beginner section
- Many users misunderstand how it works
  - To work properly, both a `RequestAuthentication` and an associated `AuthorizationPolicy` must be used
  - Only having a `RequestAuthentication` is NOT sufficient as only Authentication is performed

Thank you!

@Jacob Delgado (Aspen Mesh)
https://aspenmesh.io/