Twistlock™

Is Istio the Most Next Gen Next Gen Firewall Ever?

www.twistlock.com

Contents

What even is a NGFW?

What does Istio have to do with firewalls anyway?

What does Istio do better than classic NGFWs?

What does an Istio enabled environment look like?

Twistlock

What Even is a NGFW?

The term “NGFW” is as broad, poorly defined, and vendor abused as “cloud”

Generally, NGFWs have some layer 7 awareness, some re-perimeterization / de-perimeterization capabilities

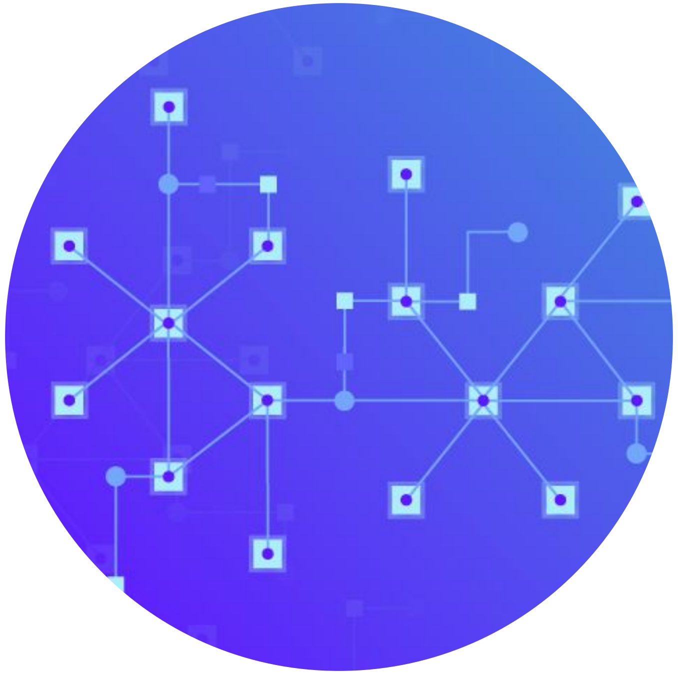
As HTTPS became the universal firewall bypass protocol, traditional L3/L4 firewalls were largely blind to relevant threats

Deperimeterization Isn’t New...

De-perimeterization as a network design concept dates back at least to 2003 with the Jericho Forum

Initially was focused on B2B collaboration scenarios and end user access, evolved into more complete scenarios like Google’s BeyondCorp

An invisible revolution - you probably just expect to work anywhere from any device



… and It’s Not Just About End User Access

It’s long been a best practice to segment datacenters into compartments… DMZs have been the almost universal minimum

Relative few have gone further than separating beyond DMZ, management, storage

Most everyone agrees that it’s *The Right Thing To Do ©*, but it’s hard to deploy a true least privilege model when everything is manual

Even harder still to operate it over time

Cloud Native Makes It Harder...

Think about your cloud native infrastructure… it's abstraction on top of abstraction, especially from a networking standpoint

Everything is ephemeral and everything is constantly changing

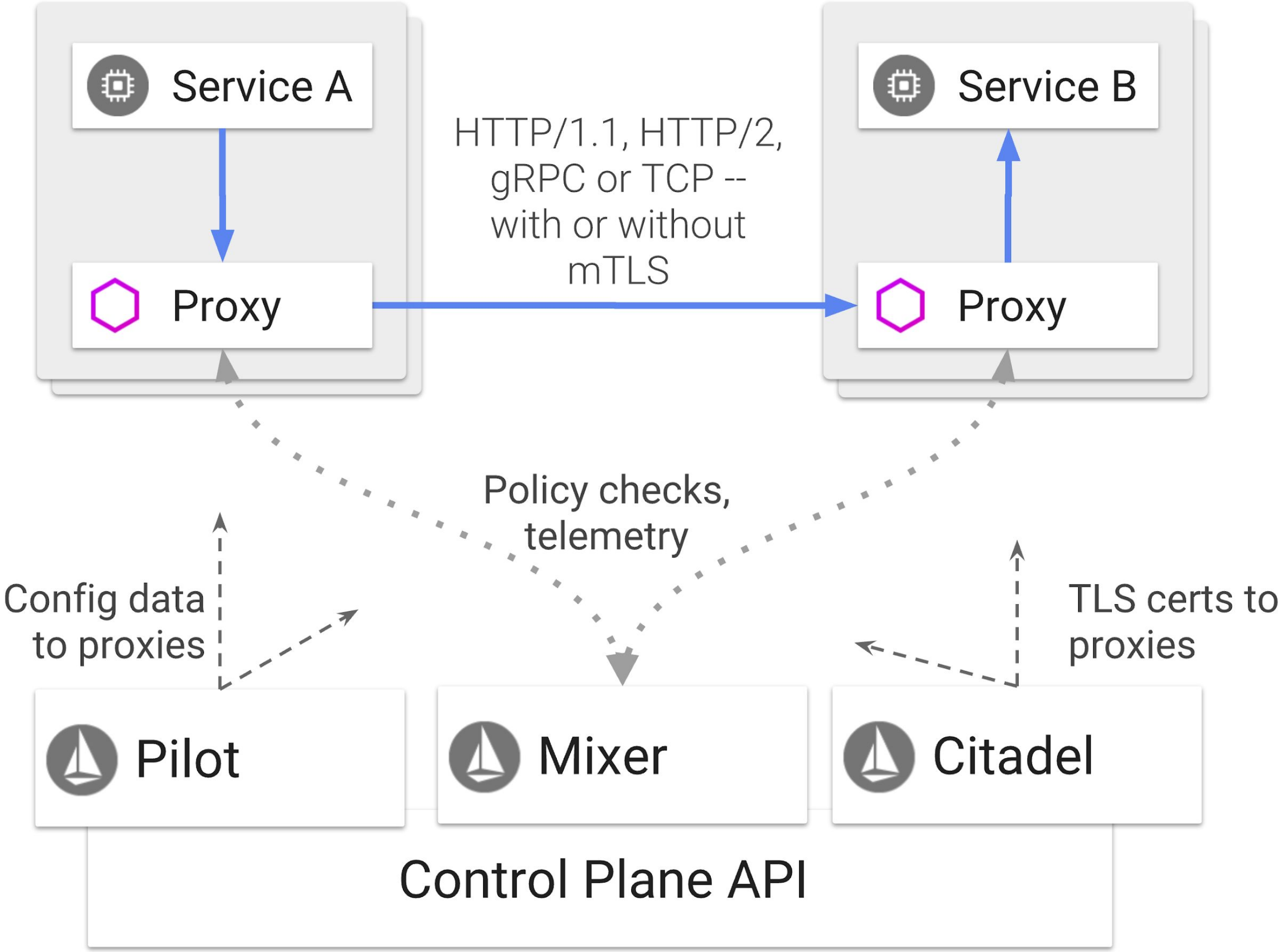
Classic segmentation approaches using VLANs or statically configured IP restrictions are impractical

…But Also Easier

Cloud native means programmable and the notion of “infrastructure as code” also applies to networking

Istio provides a new abstraction layer that separates traffic flow from virtual infrastructure to simplify traffic management, security, and observability

Not Kubernetes specific, but a natural pairing that makes easier to manage ingress and provide always on security using native Kubernetes concepts like service accounts


Service A
Proxy
HTTP/1.1, HTTP/2, gRPC or TCP -- with or without mTLS
Service B
Proxy
Policy checks, telemetry
Config data to proxies
TLS certs to proxies
Pilot
Mixer
Citadel
Control Plane API

Istio as NGFW

Deployment policy to simplistically define what services can talk to what other services

Define services and traffic - not nodes and IP addresses

Policy works *anywhere* you deploy - regardless of cloud provider or underlying hardware

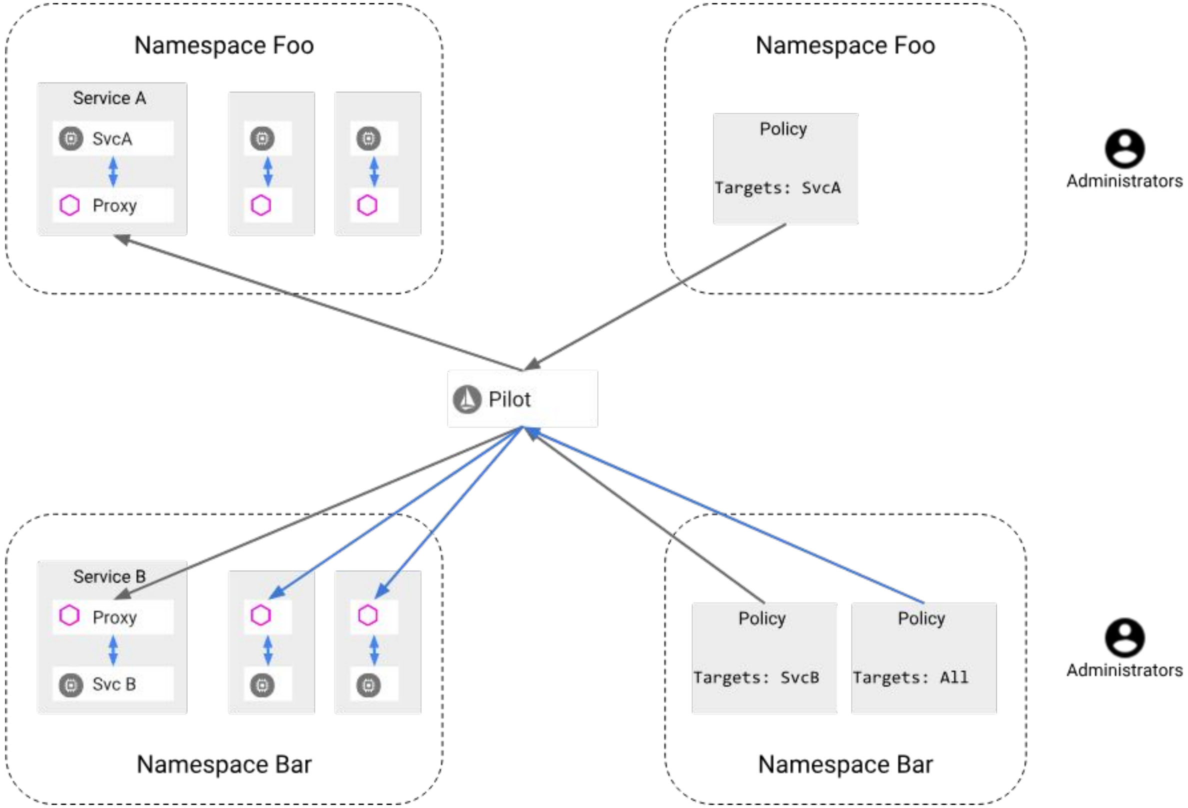
```
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRole
metadata:
  name: tester
  namespace: default
spec:
  rules:
  - services: ["test-*"]
    methods: ["*"]
  - services:
["bookstore.default.svc.cluster.local"]
    paths: ["*/reviews"]
    methods: ["GET"]
```

Authentication Policy

Determines how services identify themselves to each other

Enabled mutual TLS

Certificate lifecycle management automatically provided by Citadel



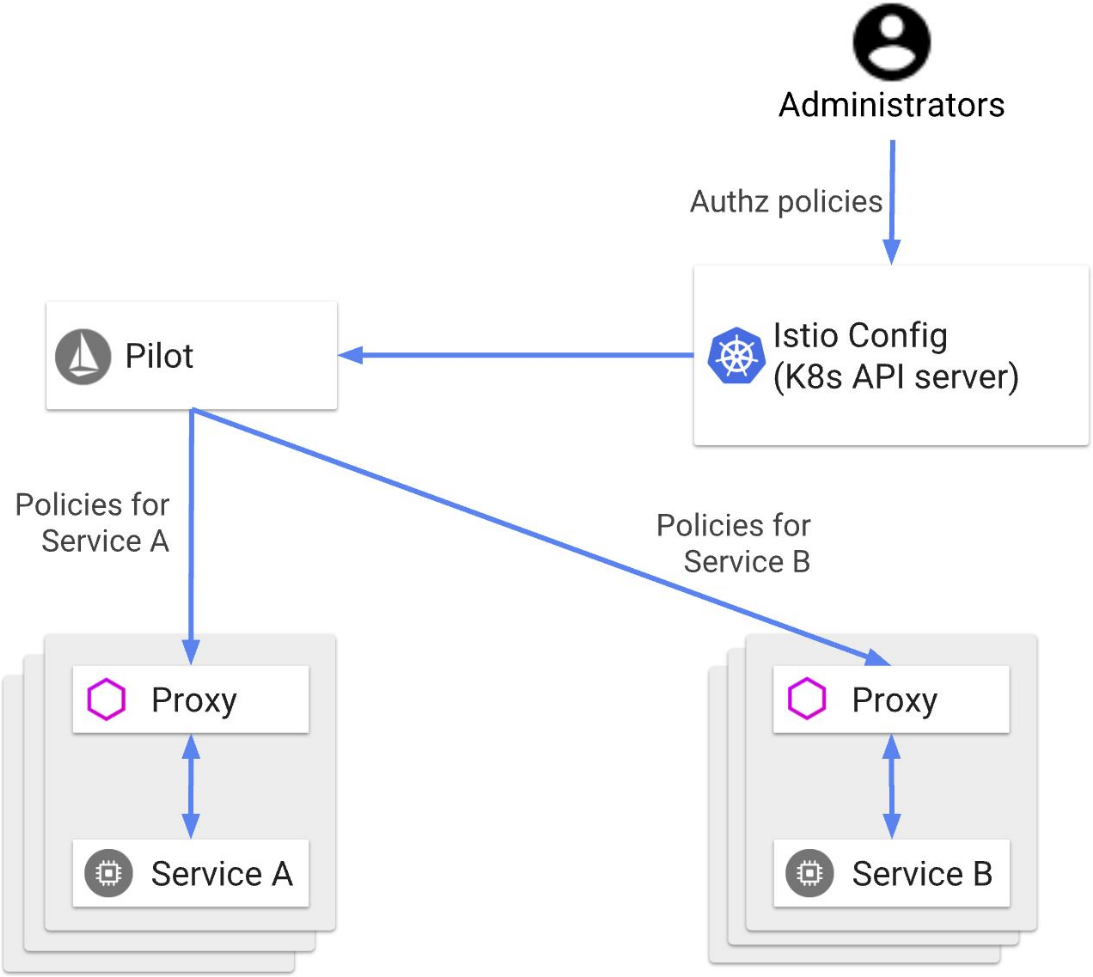
Authorization Policy

Determines what services are allowed to talk with each each other

Can be namespace-level, service-level, and method-level

Enabled via

```
mode: 'ON_WITH_INCLUSION'
```



Istio Authorization Architecture

ServiceRoles and ServiceRoleBindings

ServiceRole defines a group of permissions to access services.

ServiceRoleBinding grants a ServiceRole to particular subjects, such as a user, a group, or a service.

From a traditional NGFW perspective, analogous to creating VLANs and allowing specific interconnection paths (vlan-dmz can only talk to vlan-app)

Who, What, Which

The combination of ServiceRole and ServiceRoleBinding specifies: who is allowed to do what under which conditions

Who is the subjects section in ServiceRoleBinding

What is the permissions section in ServiceRole

Which conditions refers to the conditions section you can specify with the Istio attributes in either ServiceRole or ServiceRoleBinding

Example AuthZ Policy

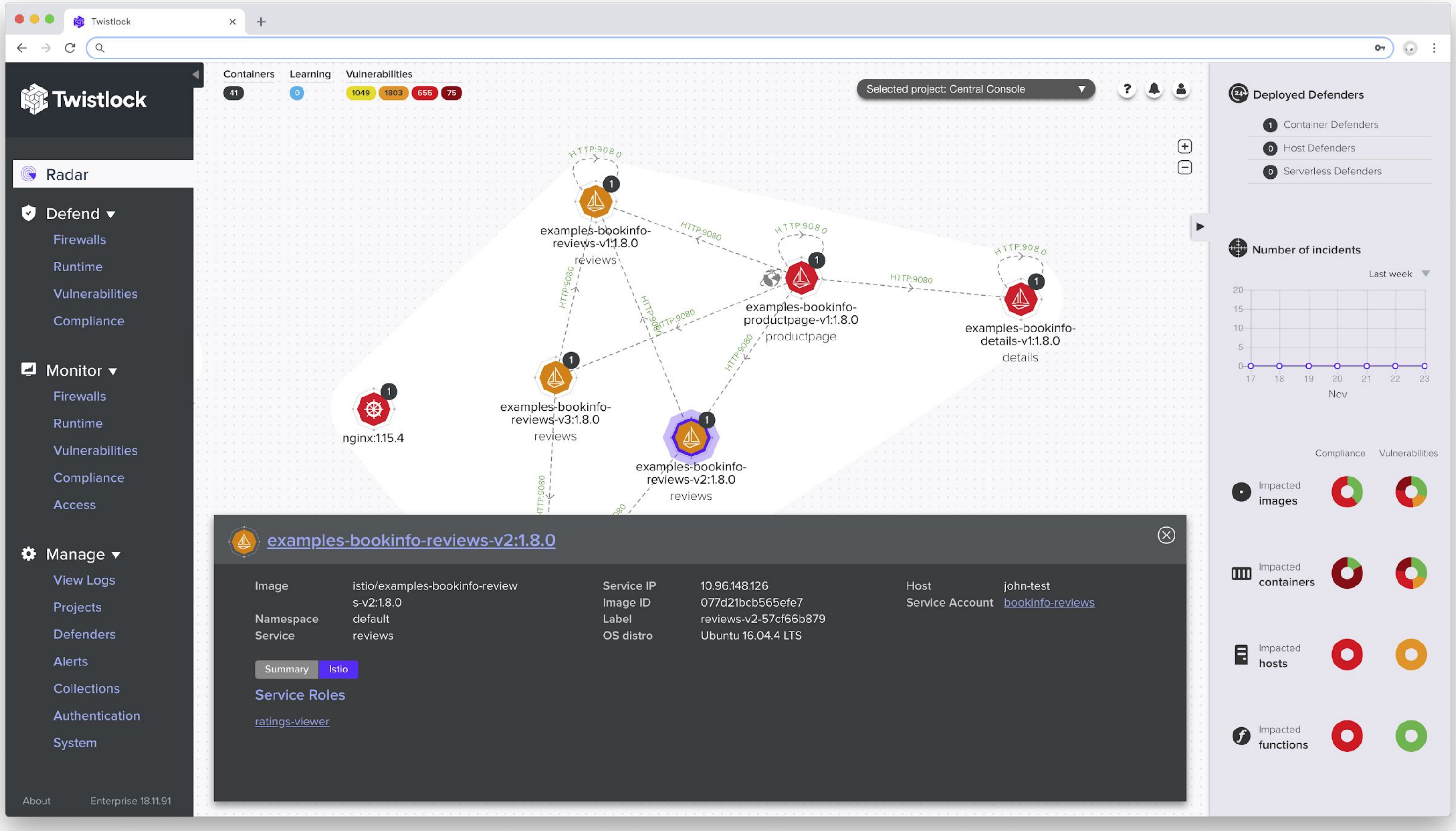
Products-viewer role, which has read, "GET" and "HEAD", access to the service products.default.svc.cluster.local in the default namespace

```yaml
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRole
metadata:
  name: products-viewer
  namespace: default
spec:
  rules:
  - services: ["products.default.svc.cluster.local"]
    methods: ["GET", "HEAD"]
```

Completely API Enabled

Can discover and model the entire topology

Build a real, live ‘Google Maps’ of your environments



examples-bookinfo-reviews-v2:1.8.0

Image	istio/examples-bookinfo-reviews-v2:1.8.0	Service IP	10.96.148.126	Host	john-test
		Image ID	077d21bcb565efe7	Service Account	bookinfo-reviews
Namespace	default	Label	reviews-v2-57cf66b879		
Service	reviews	OS distro	Ubuntu 16.04.4 LTS		

Summary | Istio

Service Roles

ratings-viewer

Bringing It Together

Microsegmentation is good but hard with traditional tools

Cloud native makes it both harder and *potentially* easier

Istio provides some key NGFW-like capabilities: encryption and RBAC, abstracted from the underlying infrastructure

“NGFW as code”

Can build immersive visualizations that provide better security understanding

Credits

John Morello
@morellonet

Excellent Istio Docs
https://istio.io/docs/

www.twistlock.com

Twistlock