RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: **CSV-W02**

Kubernetes
Practical Attack and Defense

Jay Beale

CTO
InGuardians
@jaybeale and **@inguardians**

#RSAC

What Will You See Today?

- Attack Surface of a Kubernetes Cluster
- Demonstration of a Full Attack Path on Kubernetes
- Defense Demonstration to Break the Attack Path
- Counter-Attack to Break the Defense Demo
- Demo of an Attack Leveraging Cloud APIs to Defeat Kubernetes
- Demonstration of Defenses for Cloud API Attacks
- Discussion of Additional Defenses
- Release of New Versions of Two Open Source Tools

What Can We Attack on a Master Node?

- API Server:
  - Receives requests and and serves as first point of contact
- etcd Server
  - Stores the state of the cluster, alerts subscribed components
- Controller Manager
  - Runs control loops to bring state to parity with etcd
- Scheduler
  - Bin-packs containers onto nodes
- Kube-DNS
  - Gives every requested network endpoint a name

What Can We Attack on a Worker Node?

- Kubelet: Ties the node back to the master components
- Container Runtime (e.g. Docker): Instruct the Linux kernel to create containers
- Host Operating System
  - Filesystem
  - Network
  - Kernel
- Workloads: Containers on the system
- Kube-Proxy: forwards traffic to each pod in a load-balanced network service

Demonstration

Attacking Bust-a-Kube CI/CD Scenario

Summary: Attack Bust-a-Kube CI/CD Scenario

- We achieved remote code execution via the front-end application.
- We explored that application’s service account privileges.
- Attacked other applications on the cluster to move laterally.
- Gained remote code execution in a microservice container.
- Attacked and gained remote code execution in another microservice.
- Used the final container’s privileges to compromise the entire cluster.

Demonstration

Defending the Bust-a-Kube CI/CD Scenario

Summary: Defending the Bust-a-Kube CI/CD Scenario

- We forced every non-control plane pod in the cluster to run with an AppArmor profile, via a pod security policy (PSP).
- We used a volume whitelist PSP to block an attack.
- Arms-race style: we ran a PVC-based attack to evade the PSP.
- Counter-defense: break the evasion with root capability limits.

Demonstration

Attacking a Kubernetes Cluster via its Public Cloud Provider

Summary: Attacking Kubernetes via Its Cloud Provider

- Gain remote code execution in a front end application
- Access the metadata API to gain public cloud credentials
- Abuse the storage API to gain full administrative access to the cluster

Demonstration

Defending the Cluster from Cloud API-based Attacks

Summary: Defending K8S Against Cloud API Attacks

- We deployed workload identity, which gives each pod in the cluster a lesser GCP service account than the nodes.
  - This mapping from Kubernetes service accounts to Cloud provider IAM accounts can happen via a number of cloud features and OS software.
- We configured the pod service accounts for little or no cloud API privilege.

Additional Defenses (ToC)

- Seccomp System Call Whitelists
- Read-only Root Filesystems
- Service Meshes

Seccomp System Call Filtering

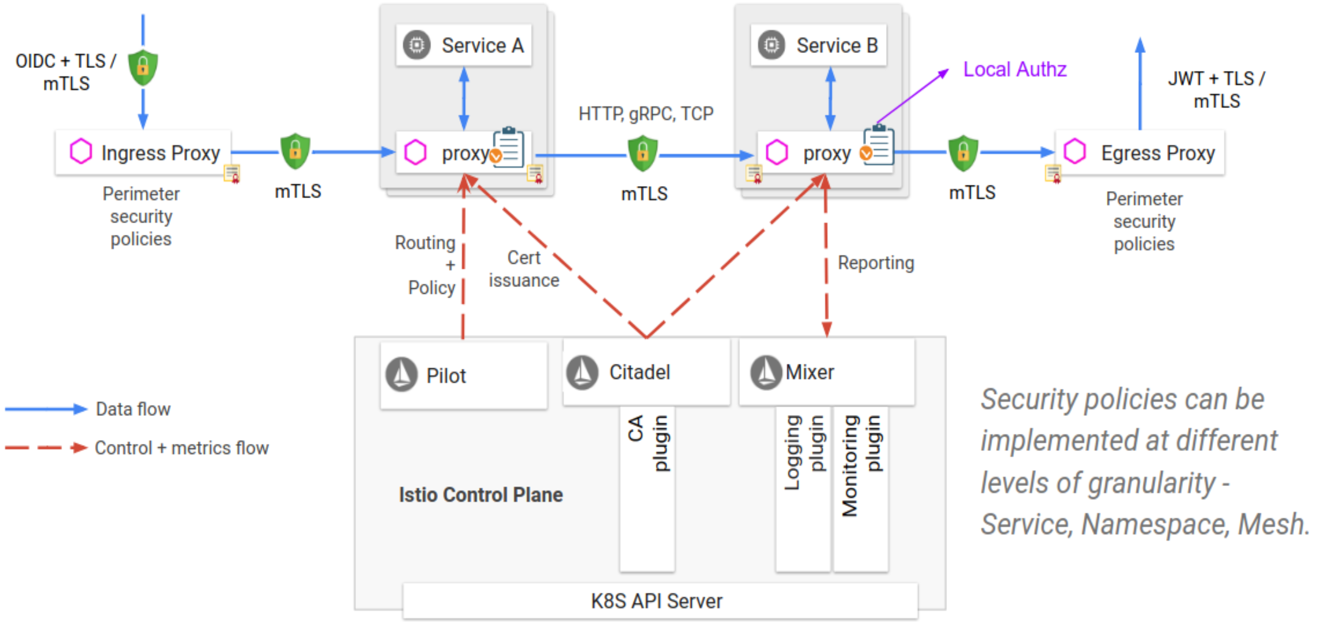
- Filtering system calls (syscalls) with seccomp has two purposes:
    - Restrict a compromised program's behavior to the system calls in its profile
    - Reduce the kernel’s attack surface
- Generate the syscall list with strace, then tell Docker or Kubernetes to confine the pod to the known list.

Read-only Root Filesystems

- Microservices lend themselves to this design pattern
- Shore up the need for writeable or persistent storage via PersistentVolumes

Service Meshes

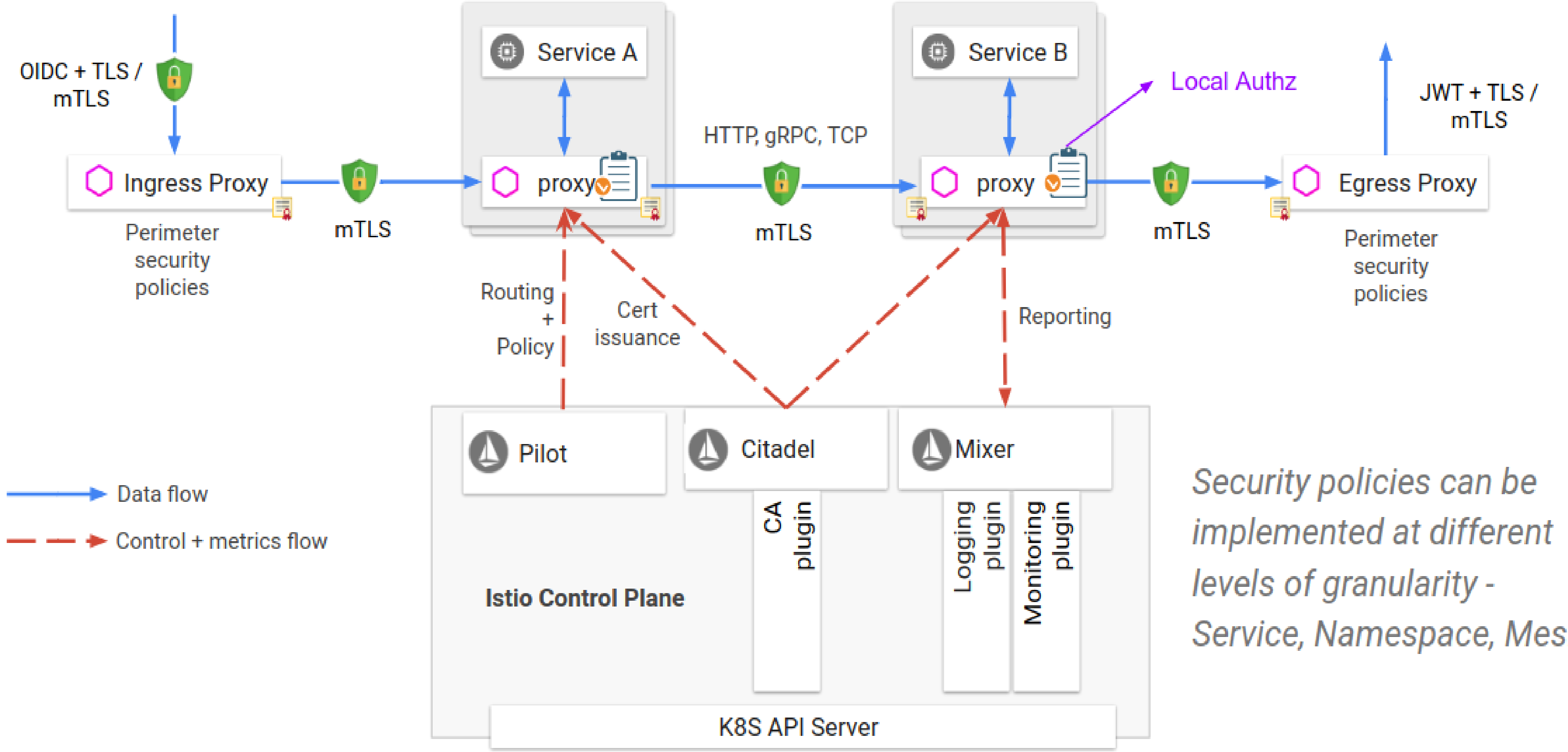
- Strong centralized control over network flow and encryption
- Accomplished via sidecar containers in every pod



Security policies can be implemented at different levels of granularity - Service, Namespace, Mesh.

(Larger version of this on next slide)

Service Mesh Example: Istio



Security policies can be implemented at different levels of granularity - Service, Namespace, Mesh.

Peirates

- The attacks here have been manual.
- We develop an open source tool: Peirates.
- Let’s see some demos!
- You can use Peirates:

  https://www.inguardians.com/peirates/
- You can help develop Peirates!

  https://github.com/inguardians/peirates

Bust-a-Kube

- You can get the same cluster that we've used in this talk's demos!
- Called Bust-a-Kube, it's an open source project.
- We use Bust-a-Kube to teach and help people train themselves on Kubernetes attack and defense.
- Download it here:

https://www.bustakube.com

Apply: Check Yourself Before Someone Wrecks Yourself

- Audit Your Authorization
  - Kubernetes RBAC
  - Cloud Roles (IAM)
- Review Your Network Controls
  - Kubernetes Network Policies
  - Service Meshes
- Contain Your Workloads
  - Pod Security Policies
  - OPA/Gatekeeper

- Upgrade Your Cluster Often
  - Kubernetes releases every 3 months
  - Clusters hit EOL by 1 year
- Apply Miscellaneous Hardening
  - CIS Benchmark: use a subet of items
  - Kube-bench will audit against this
- Pay Attention to Image Safety
  - Vulnerability scanning and mgmt.
  - Learn about software supply chain

RSAConference2020

Reference Materials

Reference: Types of Attacks via the API Server

- Ask the API Server to:
  - stage or modify containers
  - allow us to MitM network traffic
  - run commands in containers we don't own
- Ask the Kubelet to:
  - run commands in containers we don't own
  - display details of all workloads running in the cluster

Reference: Attack Types Added by Cloud APIs

- Interact with the Cloud Provider
  - Obtain node's credentials from the Metadata API
  - Gain Kubernetes authentication tokens from cloud storage buckets
  - Modify or create compute instances
  - Modify or duplicate storage
  - Interact with any API that the node can