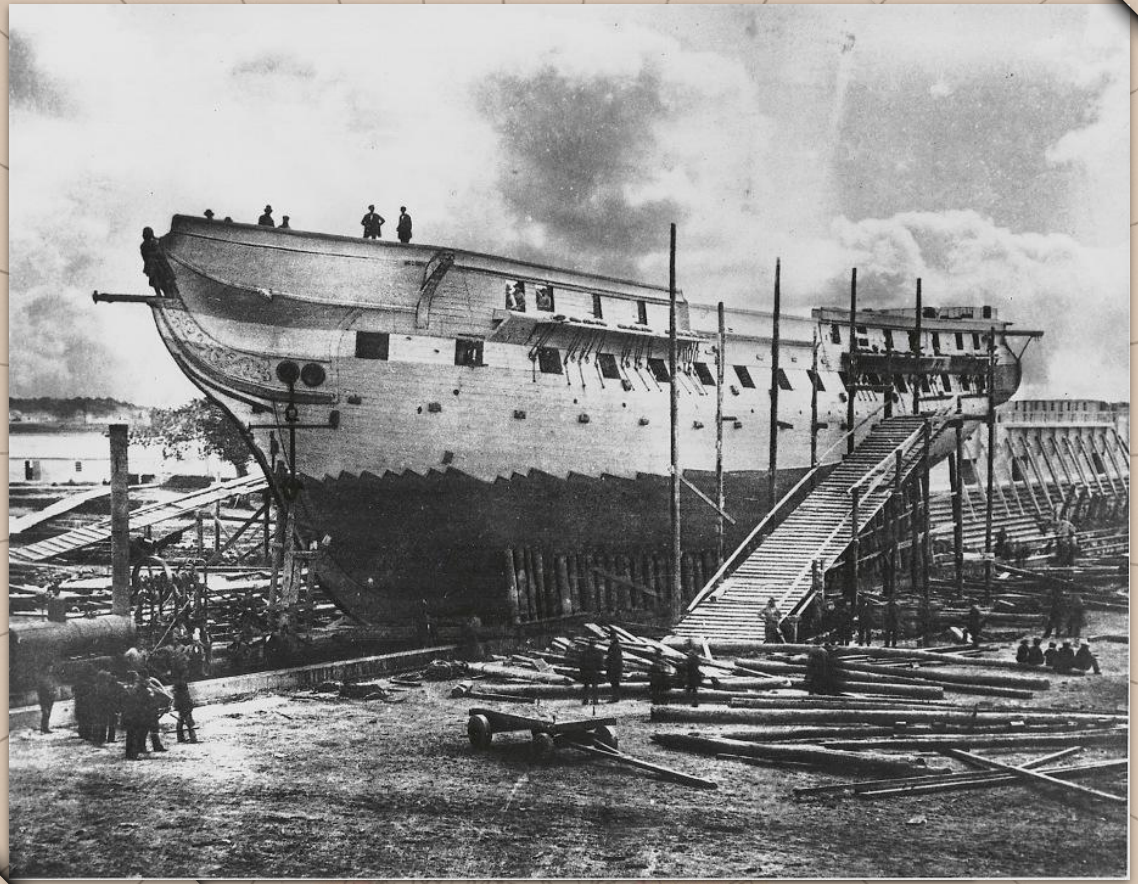
Breaking down the ship

How Kubernetes is put together and how you can take it apart

BSIDES PORTLAND



Ahoy!

I am **Alex Ivkin,** Director of Solutions at Eclypsium, a US firmware security company, from Portland, Oregon.

I do secure deployments in (in)secure environments, including container orchestration, application security, and firmware security.

The modern application stack


Application


Orchestrator


Container


OS/Kernel


Platform

The modern application stack

Application

Orchestrator

Container

OS/Kernel

Platform

The modern application stack

Application

Orchestrator

Container

OS/Kernel

Platform

HORRIBLE BUT SOMEBODY ELSE NEEDS TO CARE

The modern application stack

Application

Orchestrator

Container

OS/Kernel

Platform

HORRIBLE BUT SOMEBODY ELSE NEEDS TO CARE

The modern application stack

Application

Master - API, etcd, Controller, Scheduler, Manager

Worker - Namespaces, mounts, cgroups, syscall filters, caps...

OS/Kernel

THIS IS BAD

Platform

HORRIBLE BUT SOMEBODY ELSE NEEDS TO CARE

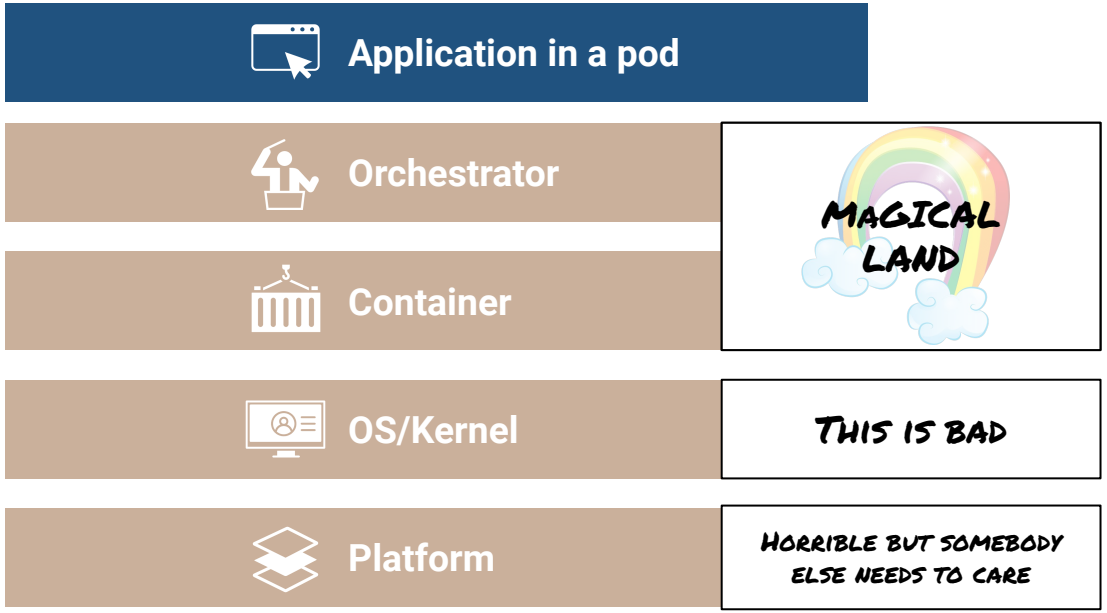
Arm! Drop down, fire!

Host device mount, privileged container escapes, kernel triggers via sysfs callbacks, CAP_SYS_ADMIN – cgroup notify on release escape, CAP_SYS_Module – Load Kernel Module

- **BSides PDX 2018** - Containers with windows, elevators and backdoors
- **OWASP PDX Training 2019** - Container Security

https://github.com/alexivkin/containerpwn

The modern application stack



Where containers will not help

If your app is bad, it won't fix it

- Injection/Insecure Deserialization/RCE
- Improper input validation
- Use-after-free, TOCTOU, overflow
- XSS, CSRF, SSRF

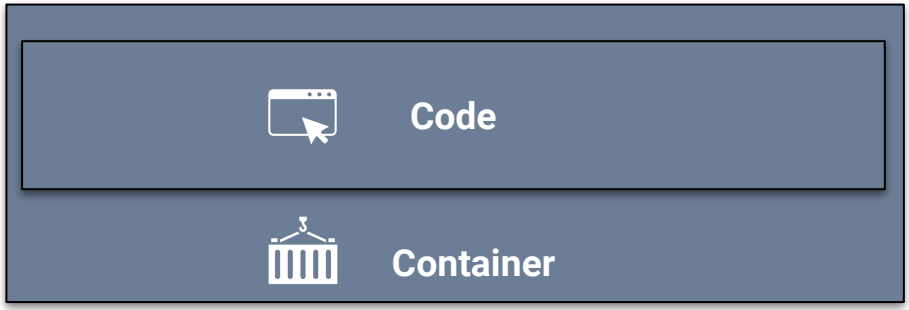
The modern application stack

Application	THIS IS BAD
Orchestrator	
Container	
OS/Kernel	THIS IS BAD
Platform	HORRIBLE BUT SOMEBODY ELSE NEEDS TO CARE

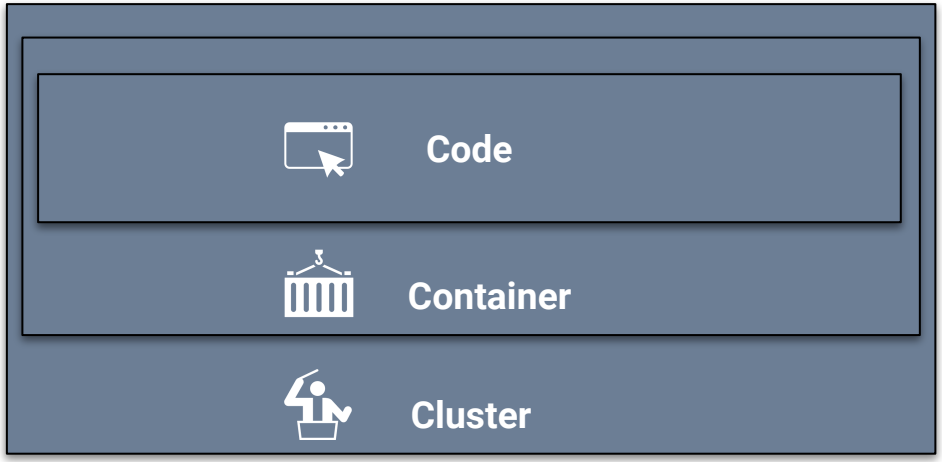
The modern matreshka

Code

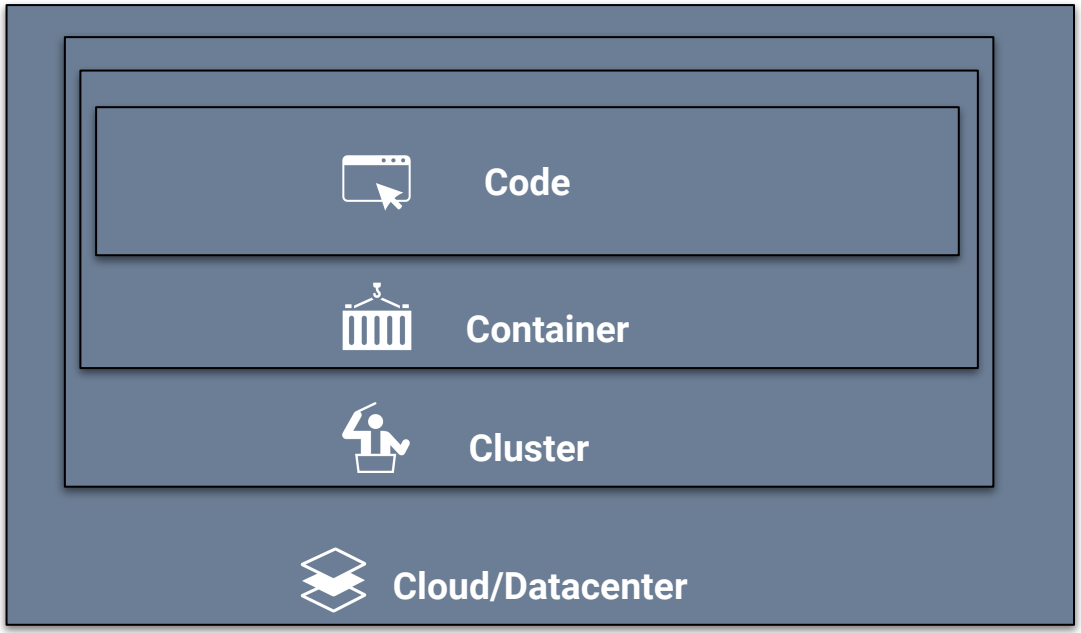
The modern matreshka



The modern matreshka

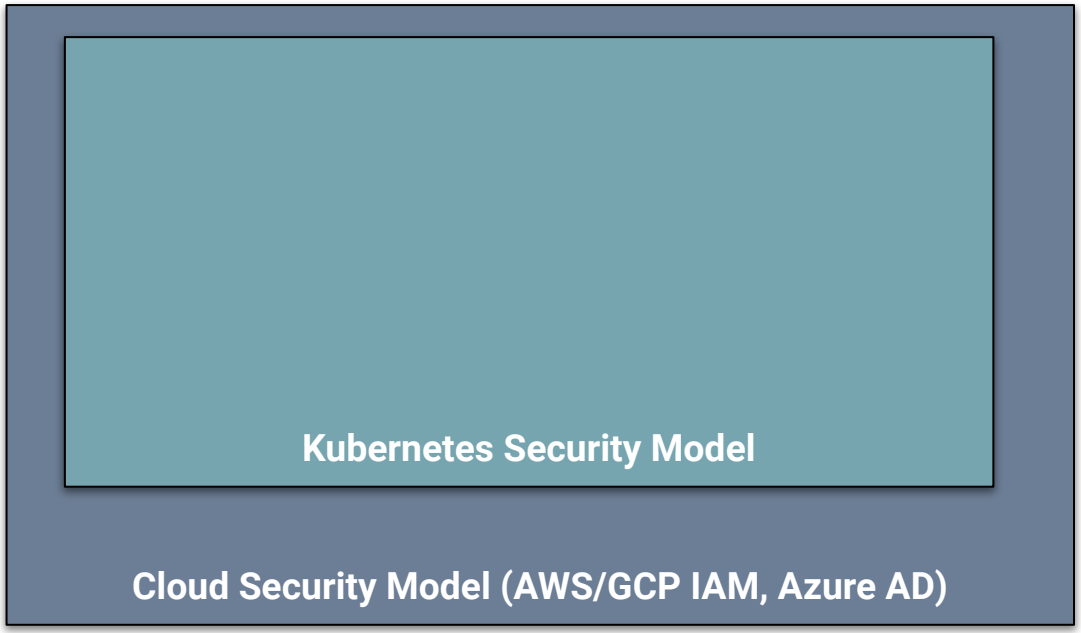


The modern matreshka

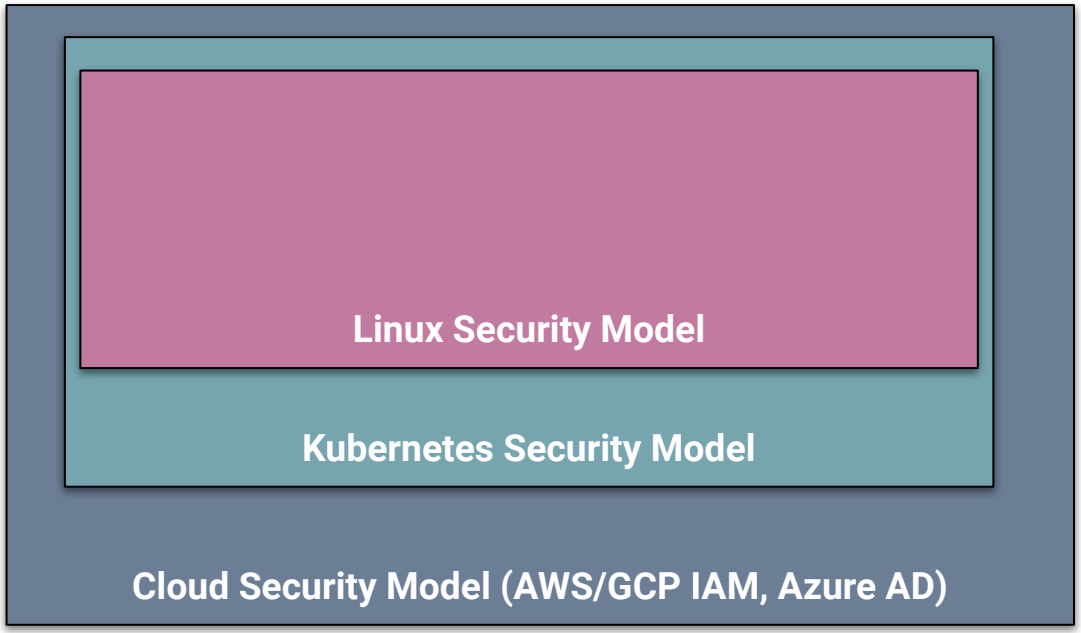


The modern matreshka

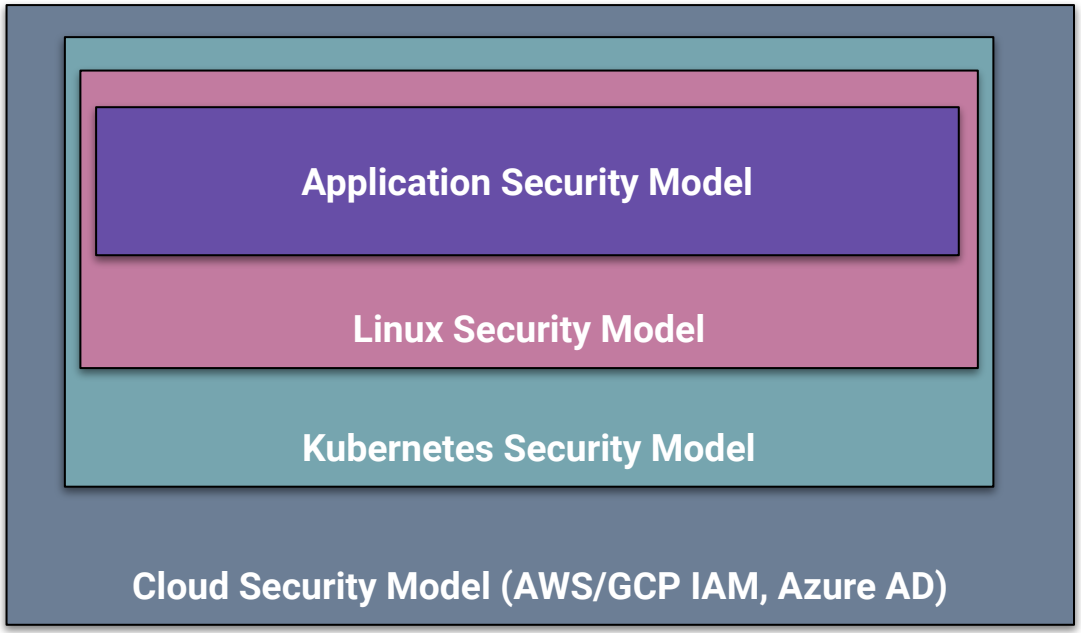
The modern matreshka



The modern matreshka



The modern matreshka



Kubernetes Security Model

- User Accounts
- Service Accounts
- RBAC
- Namespaces
- Policies

Kubernetes Auth/Authz

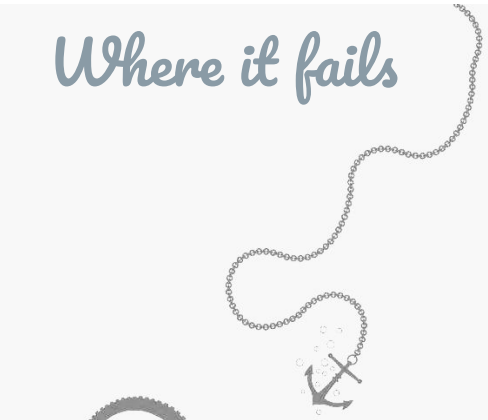
- Mutual TLS between API server and clients
- Identification through TLS certs
- Roles and role-bindings to identities
- Enforcement through API plugins

One security audit found Kubernetes security model reasonably good

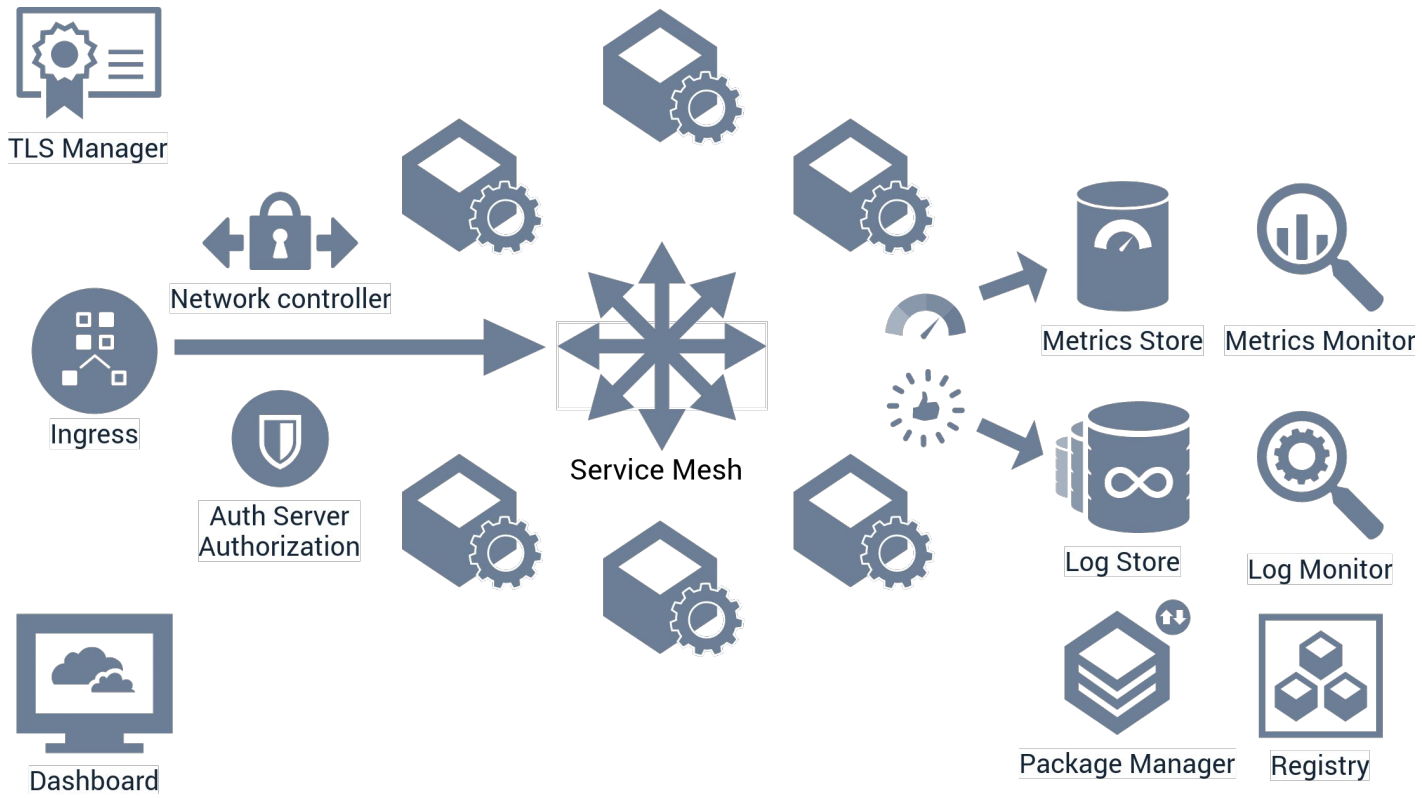
- CVE-2017-1002101 - Accessing files outside of a subpath volume mount via symlinks
- CVE-2017-1002102 - Arbitrary file deletion in host filesystem
- CVE-2018-1002105 - Privesc via API request crafting. PoC 1 source, PoC 2 source
- CVE-2018-18264 - Wide open dashboard
- CVE-2019-9946 - CNI iptables misuse
- CVE-2019-5736 - A container escape by overwriting runc binary. PoC 1, PoC 2
- CVE-2019-11245 - Kubelet security regression
- CVE-2019-11246 - Container tar binary can mess with the client that's running kubectl cp
- CVE-2019-11247 - modifying CRDs across the cluster
- CVE-2019-11251 - kubectl cp dir traversal via symlinks
- CVE-2019-11253 - denial of service vulnerability from malicious YAML or JSON
- CVE-2019-16276 - API Authenticating Proxy bypass via HTTP Protocol Violation in Go's net/http Library
- CVE-2019-1002101 - Dir traversal via copy
- CVE-2020-8551 - Kubelet DoS via /etc/hosts stuffing
- CVE-2020-8552 - API server DoS OOM
- CVE-2020-8555 - Half-Blind SSRF in kube-controller-manager. Full writeup
- CVE-2020-13597, CVE-2020-10749, CVE-2020-13401 - MitM attacks via IPv6 rogue router for CNI like Calico, Flannel, WaveNet
- CVE-2020-8557 - Node disk DOS by writing to container /etc/hosts
- CVE-2020-8558 - net.ipv4.conf.all.route_localnet=1 setting allows for neighboring hosts to bypass localhost boundary. PoC source
- CVE-2020-8559 - Privilege escalation from compromised node to cluster. PoC source

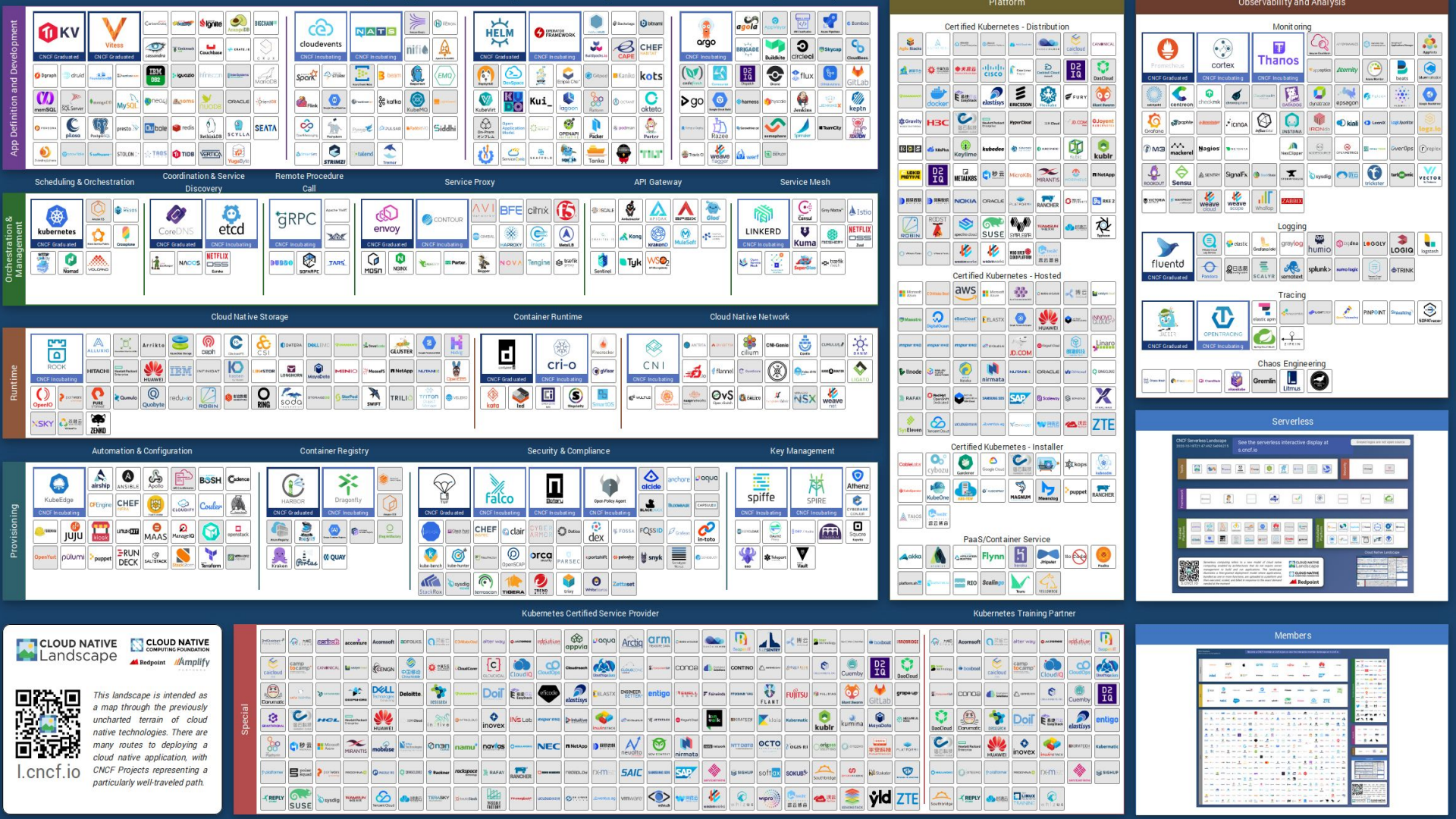
Where it fails

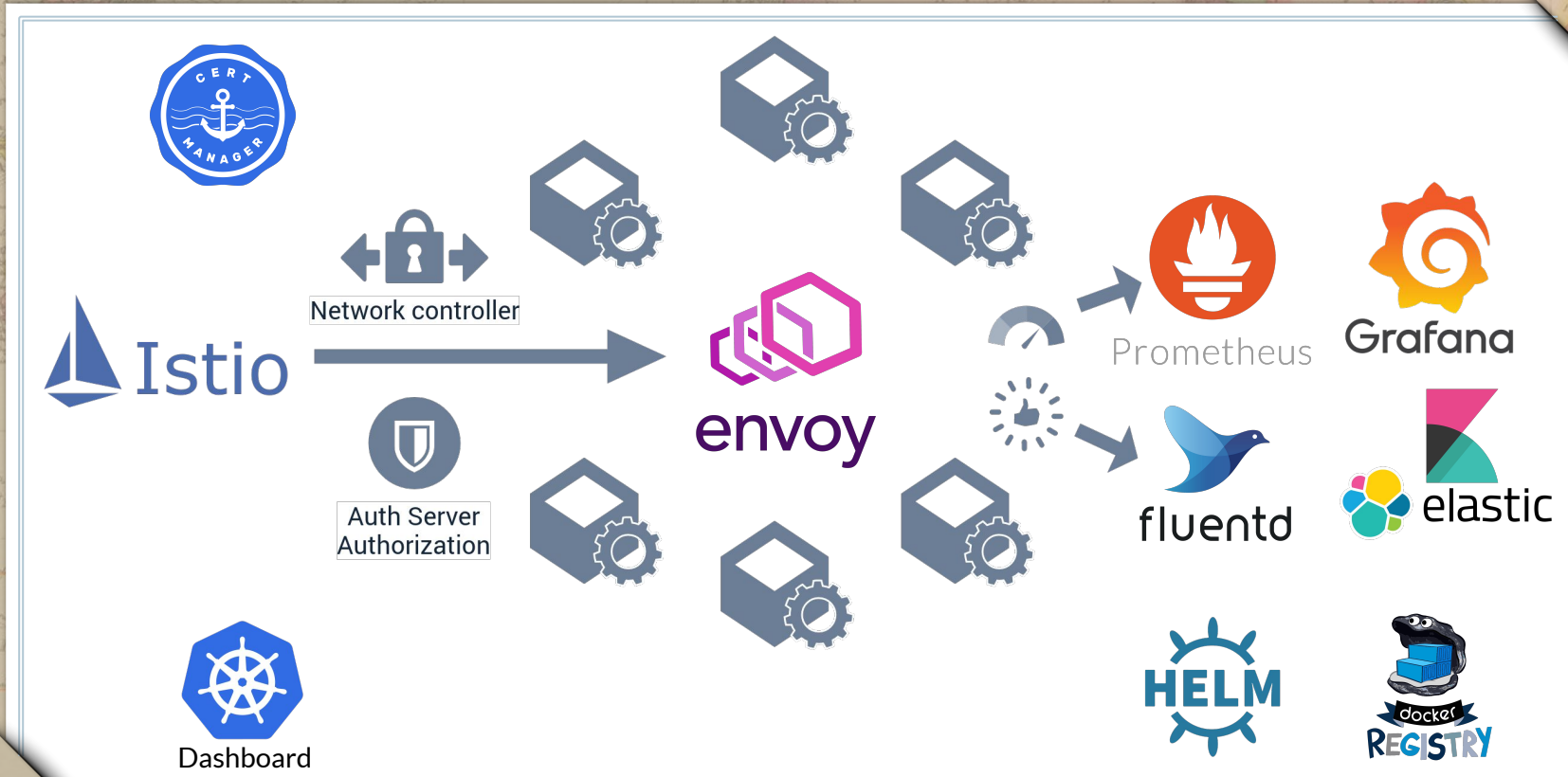
Operate your app on a cluster

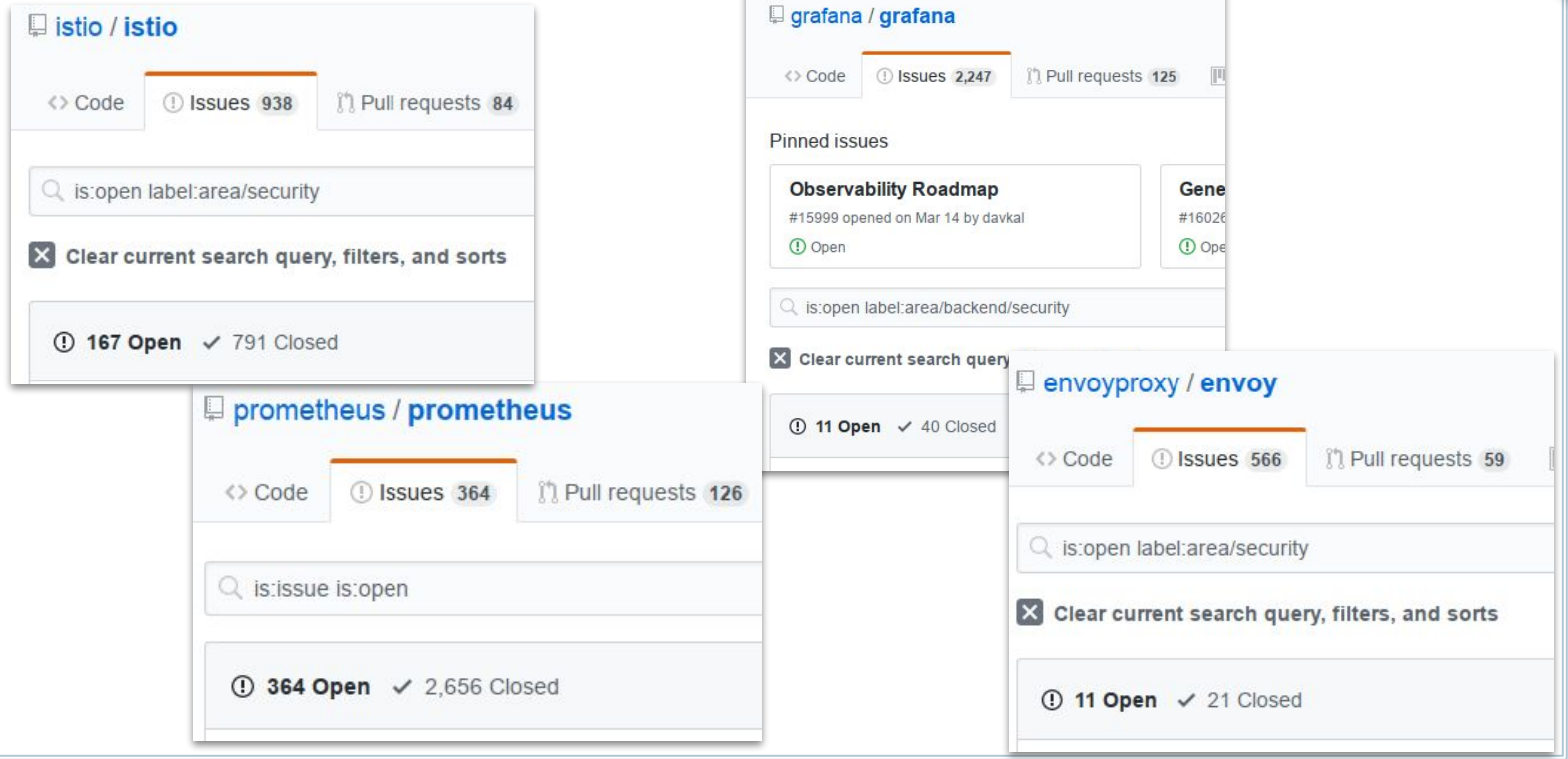


Maintain your cluster


TLS Manager
Network controller
Ingress
Auth Server
Authorization
Service Mesh
Metrics Store
Metrics Monitor
Log Store
Log Monitor
Package Manager
Registry
Dashboard

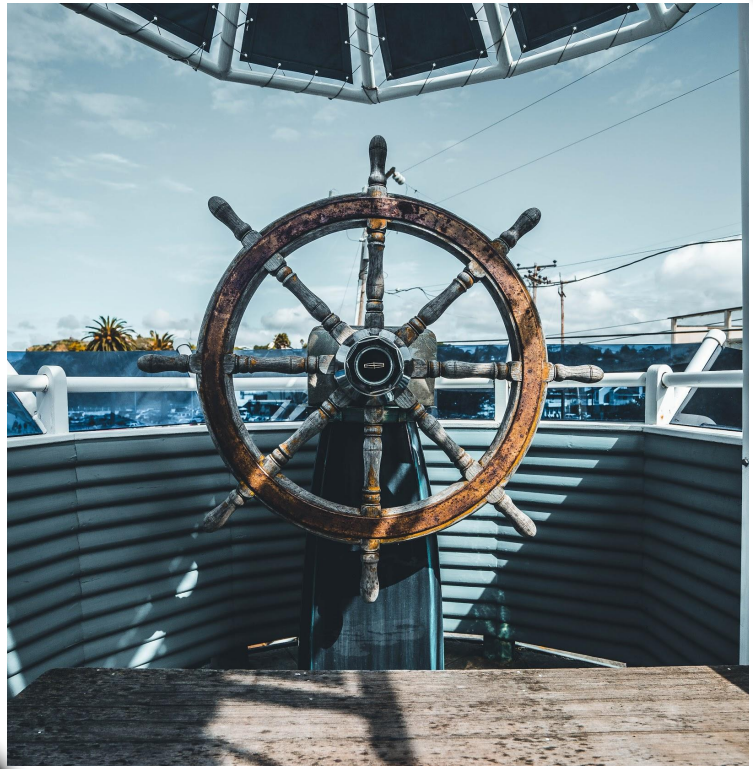

App Definition and Development
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Orchestration & Management
Cloud Native Storage
Container Runtime
Cloud Native Network
Runtime
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Provisioning
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Serverless
Kubernetes Certified Service Provider
Kubernetes Training Partner
Special
Members
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
l.cncf.io


CERT MANAGER
Network controller
Istio
Auth Server
Authorization
envoy
Prometheus
Grafana
fluentd
elastic
Dashboard
HELM
docker
REGISTRY

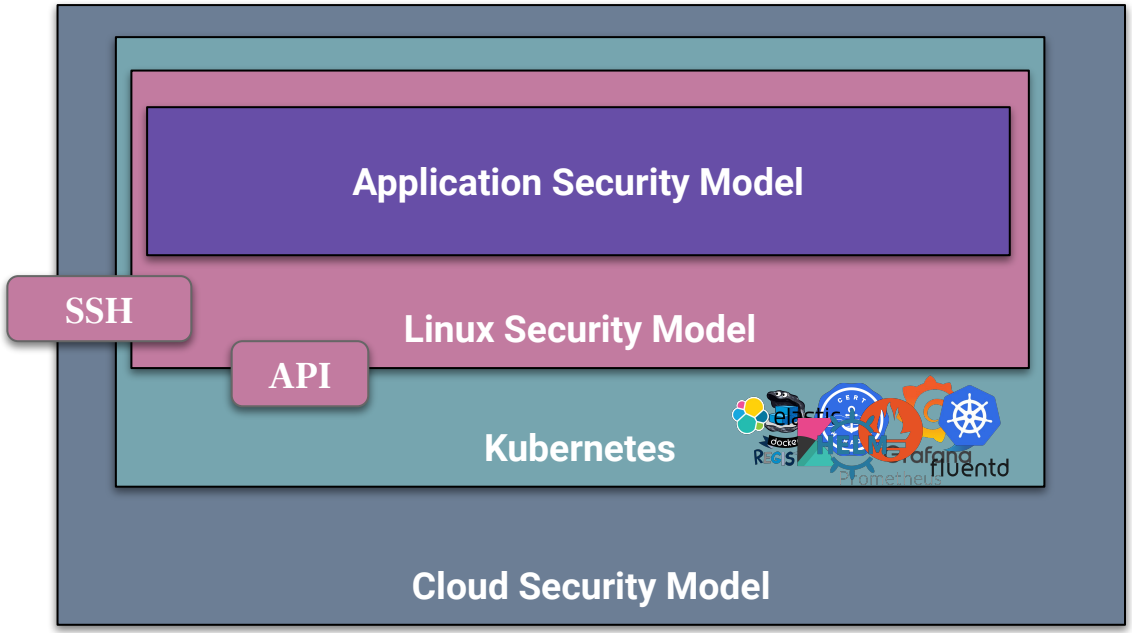

istio / istio
Code
Issues 938
Pull requests 84
is:open label:area/security
Clear current search query, filters, and sorts
167 Open
791 Closed
grafana / grafana
Code
Issues 2,247
Pull requests 125
Pinned issues
Observability Roadmap
#15999 opened on Mar 14 by davkal
Open
is:open label:area/backend/security
Clear current search query
11 Open
40 Closed
prometheus / prometheus
Code
Issues 364
Pull requests 126
is:issue is:open
364 Open
2,656 Closed
envoyproxy / envoy
Code
Issues 566
Pull requests 59
is:open label:area/security
Clear current search query, filters, and sorts
11 Open
21 Closed

So what?

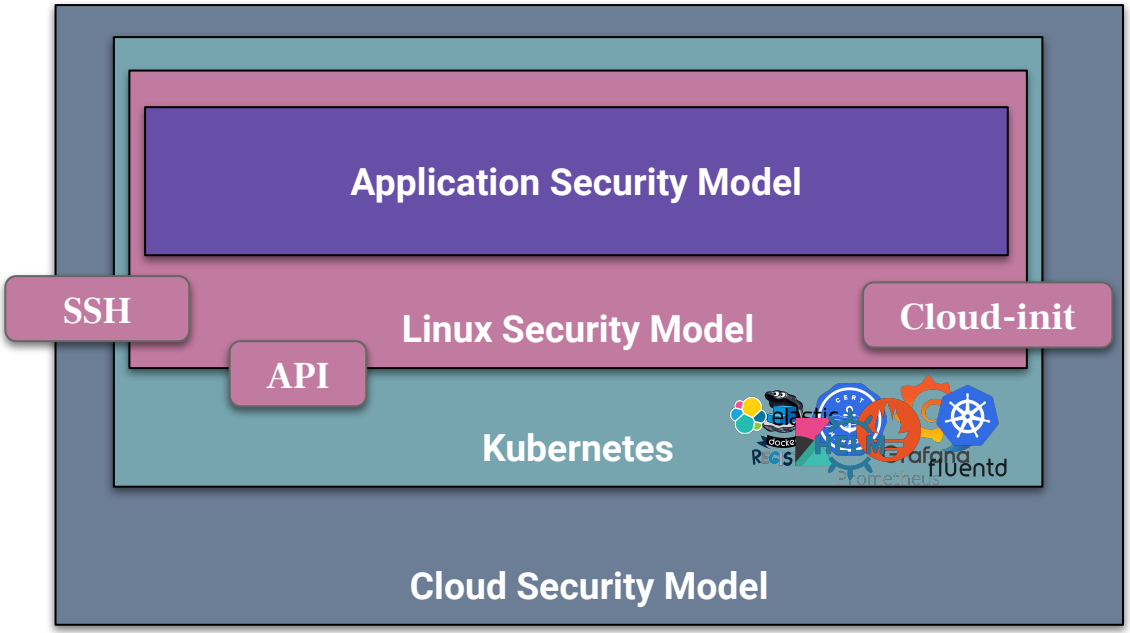
- **Service mesh** - Sidecar containers (third party processes in your app's linux namespaces)
- **Ingress** - change iptables
- **Deployment tools** - pods with rights to add/remove/configure RBAC/Pods/Services/Gateways
- **Registries** - master copy for every container process running in the cluster
- **TLS managers** - CA and private keys for transport security and API identification



The modern matreshka



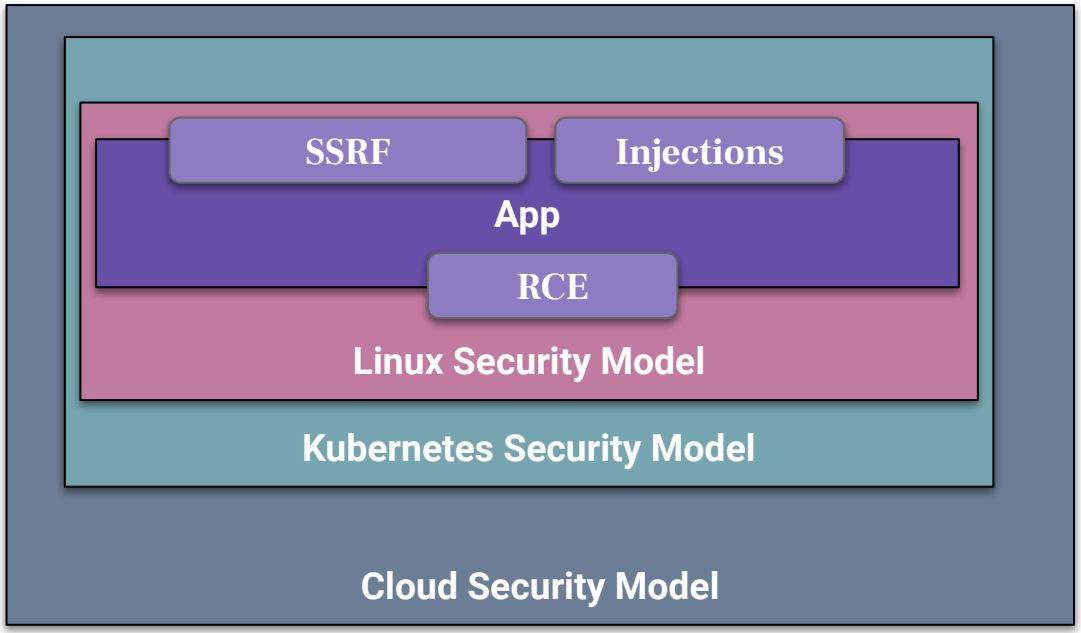
The modern matreshka




NATIONAL GEOGRAPHIC
Photograph by Jana Asenbrennerova
2010 NATIONAL GEOGRAPHIC PHOTOGRAPHY CONTEST
© COPYRIGHT JANA ASENBRENNEROVA. ALL RIGHTS RESERVED.

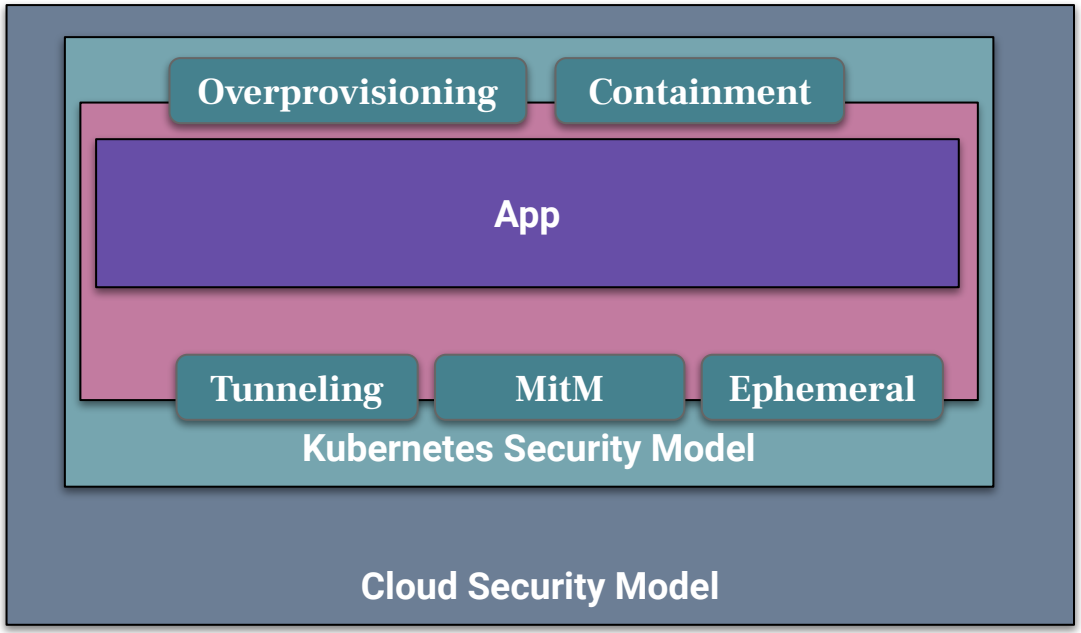
App/Linux boundary

- Burp/ZAP
- Cobalt
- Metasploit
- Your favorite scanner/fuzzer



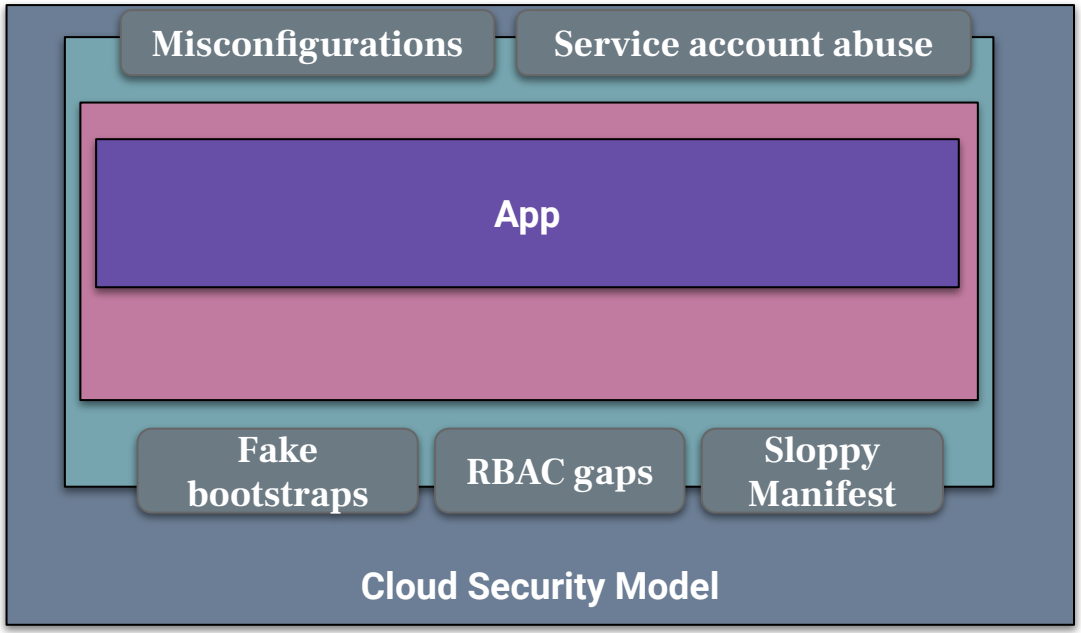
Linux/K8s junction

- BotB
- Conmanchi
- Ktunnel
- Kubetap
- Ephemeral containers



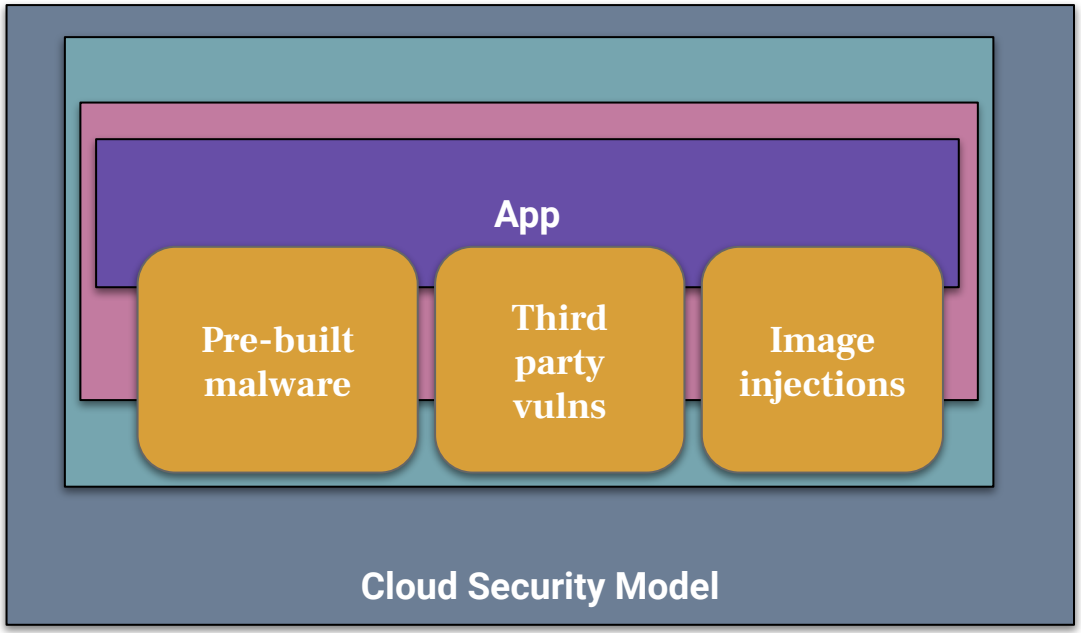
K8s/Cloud seam

- BotB and kubeletmein
- Krane and Rakkess
- Kubesec



App/Cluster gap

- Anchor, Clair, Trivy, Vuls
- Go-pillage-registries
- Dockerscan



Tis here be not jolly

Easy

- Scan your cluster with Kubiscan, Kube-bench, Managed Kubernetes Inspection Tool, Popeye
- Use simple images to start from or grab and reconstruct dockerfile and build the image yourself
- Use Pod Security Policies

https://github.com/alexivkin/docker-historian

Normal

- Build from scratch or use distroless images
- Pin images to hashes
- Upgrade master nodes and all nodes to the latest version
- Ship out K8s secrets to a vault
- SAST/DAST/SCA for images

Hardmode

- Monitor for rogue containers and processes
- Sign your images - Harbor/Grafeas/Notary
- Zero-trust (dynamic mTLS, SPIFFE/SPIRE)
- Unmix sensitive workloads
  - Container separation is weaker than hypervisors
  - Namespaces will not provide hard multitenancy
    - Use separate clusters with Gardener

Be now merry ya scallywags!

https://github.com/alexivkin

@alexivkinx

in/alexivkin

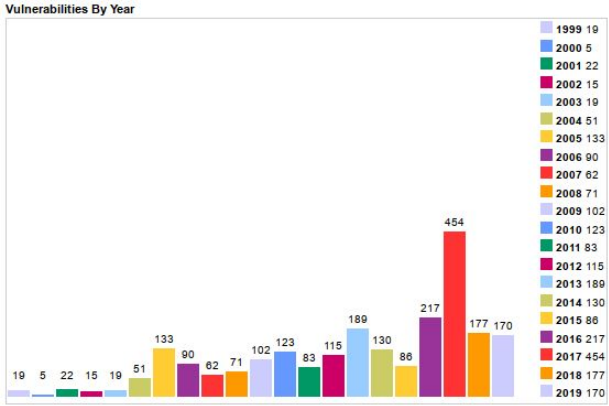
BUT

Security best practices, frameworks and standards

- CTFs and playgrounds
- CIS Benchmarks for Docker and Kubernetes
- Aqua, StackRox, SysDig, Tigera, Neuvector whitepapers
- Night time reading
  - US DoD DevSecOps reference design
  - NIST SP 800-190 - Compliance in Container Environments
  - SANS Checklist for Audit of Docker containers
- https://github.com/alexivkin/containerpwn
- https://github.com/alexivkin/kubepwn

But wait there's more!

- Docker/ContainerD Vulns
- GoLang Vulns
  - Data race breaking memory safety
- Image Vulns
  - Alpine root with no password
- Kernel Vulns



It’s gonna get more complex

Errors, not vulnerabilities, lead to a lot of breaches

Make it secure by default

Open API

2018 Tesla dashboard

2019 Monero miners on AKS

Shopify GKE Metadata exploit

Compromised clusters via kubelet API

CREDITS

- Presentation template by SlidesCarnival
- Maps in slides 2, 6 and 9 from David Rumsey Map Collection
- Photographs by Unsplash and Pexels authors.