Good Enough for the Finance Industry:

Achieving *High* Security at *Scale* with Microservices in Kubernetes

Ygrene Energy Fund, Engineers: Austin and Zach

What is Ygrene?

An why we focus on security.

*it’s also the word “energy” spelled backwards

What is this talk?

Cloud Native Security tips for any skill level.


rkt


Code & Containers

A perfect harmony… kinda

Containers are Awesome!

They solve so many problems in dev, but add many in production too.

Risk Management with Docker

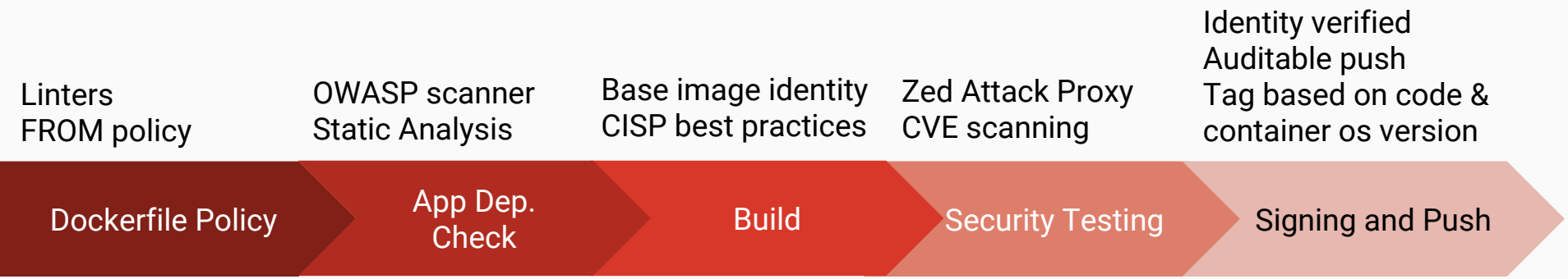
What are the risks we care about?

1. Unexpected container contents running in production
2. Data leakage into container
3. Container os and app level vulnerabilities

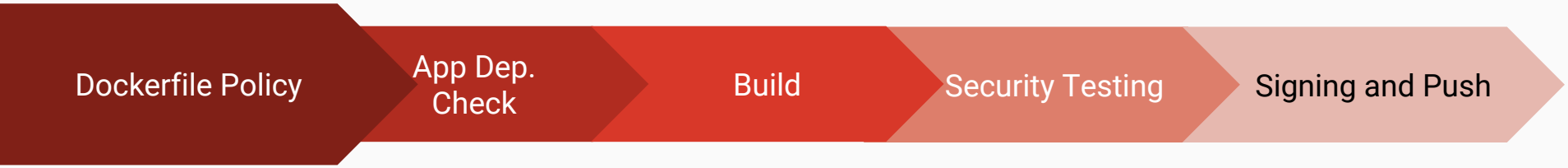
Mitigation: Building in Security

Before the container is pushed to a repo it goes through some policy enforcement.

1. Dockerfile Policy
    a. Base Image Policy
    b. Best Practices for runtime
2. Application/CVE Policy
    a. Code dependency checking
    b. Static Analysis
    c. Penetration test
    d. Linux dependency checking
3. Identity Policy
    a. How to use notary securely
    b. Base image identity verification
    c. Image sign



One pipeline to rule them all

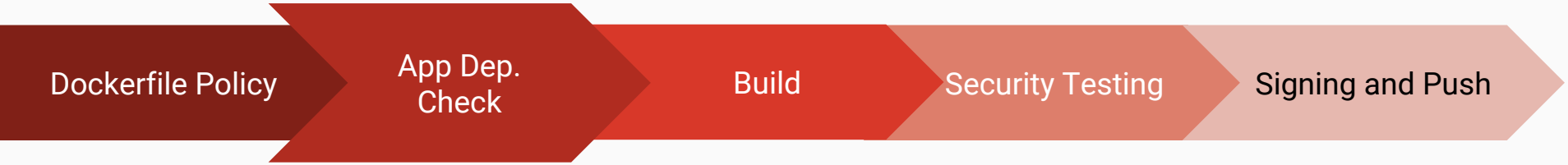


Dockerfile Policy

The goal is to achieve CISP Docker standards **and** internal organizational policies. Some tools:

https://github.com/austbot/lineage
https://github.com/hadolint/hadolint
https://github.com/replicatedhq/dockerfilelint
https://www.cisecurity.org/benchmark/docker/
https://github.com/docker/docker-bench-security

1. FROM policy - Enforcing specific base/intermediary images.
    a. Use “slim” versions, Alpine, Debian-Slim
    b. Use security focused os, Amz Linux, RedHat
    c. DON’T USE :latest
2. Best Practices
    a. Don’t run as root
    b. Check dep/package checksums
    c. Remove privileged binaries



Application/CVE Policy

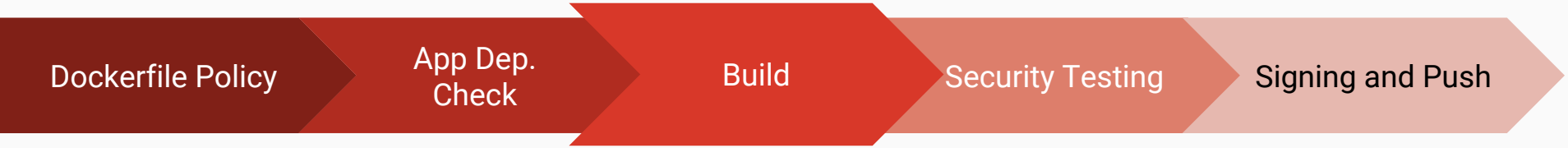
The goal is to leverage community work to weed out clear security problems.

Static Code Analysis

Dependency CVE Checking

Tools

https://www.owasp.org/index.php/OWASP_Dependency_Check
https://github.com/rubysec/bundler-audit
https://github.com/albuch/sbt-dependency-check
https://www.owasp.org/index.php/Source_Code_Analysis_Tools
https://github.com/GoASTScanner/gas



Identity Policy & Dockerfile Policy

We only allow signed base images using Docker Content Trust (Notary)

Our docker container build will validate checksums of resources needed through curl/wget

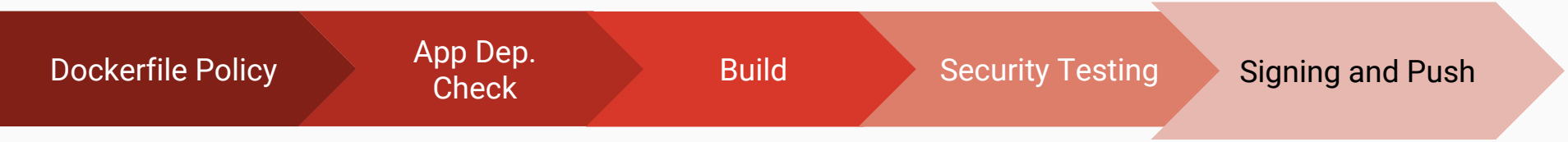
We always squash to get rid of secrets or keys that maybe needed for build but not runtime.

Application/CVE Policy

After the container is built we scan the OS for vulnerabilities. The build is rejected if there are high vulnerabilities. On the registry after push the image is scanned passively and we are notified of new CVE's affecting our containers. Then we attack the container using OWASP Zaproxy.

Tools

https://www.zaproxy.org/
https://github.com/eliasgranderubio/dagda
https://www.clamav.net/
https://coreos.com/clair/docs/latest/



Identity Policy

Developers can build images for local dev, ect… Only build servers can build images to be run in production. This helps move away from manual, but its still possible if necessary.

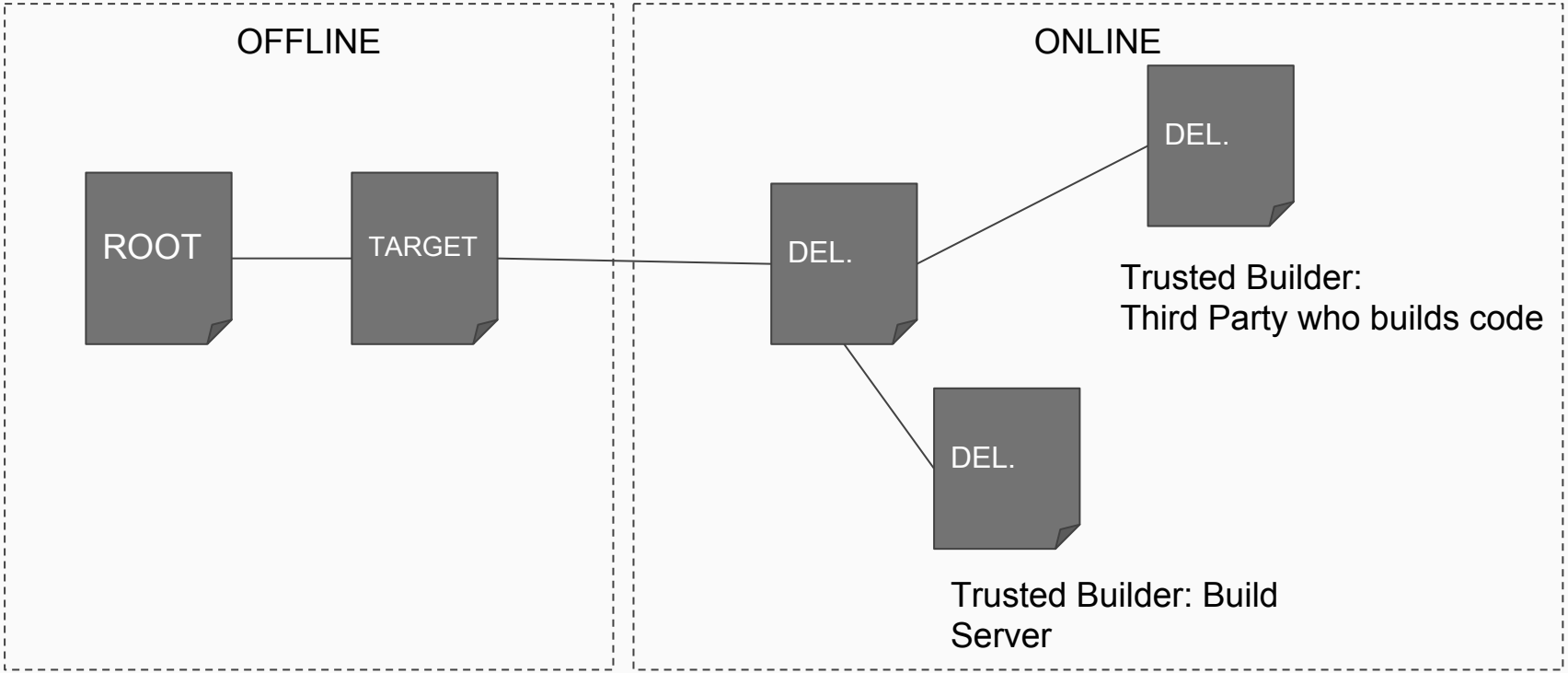
Notary Tips:

a. Never use your root key.
b. Delegations keys all the way.

Tools

https://quay.io

A good way to use Notary.



We don’t use our root key or targets key anymore. Use delegation keys, they can easily been revoked.

Cluster/Cloud*

*or co-lo, corporate datacenter...

Issues in Kubernetes for the Risk Averse

- Locus of Control
- OSS
- Attack Surface
- Same vulnerabilities as Virtualization
  - 3rd party dependency vulnerabilities (covered earlier)
  - OS Vulnerabilities (covered in containers)
  - Network vulnerabilities (kube-dns and such)
  - Auth-z, Auth-n
- Improper cluster bootstrap and setup

K8s Community Tools to Mitigate Risks

- Kubernetes' list of 1000 API Server Flags and Feature Gates
  - RBAC
    - Adjust the default service account for the namespace, it's fairly permissive
  - API Server Admission Controllers
    - Mutating/Validating
  - Audit Logging
  - Network Security Policies
  - Pod Security Policies
  - Taints and Tolerations
- Kops, Kubespray, Kubeadm...don't have security focused defaults

External Tools to Help Mitigate Risks

- Aqua Scanner’s Kube Bench project
- Dex or other tools for IAM inside the cluster
- Service Mesh tools (such as Istio, Linkerd, Conduit and others…)

Use a managed service.

GKE, AKS, EKS, Tectonic, and most of the other vendors in the showcase...

If you can’t use a managed service

- Use a tool like Kops
- Advantages:
  - Git versioning of your K8s cluster
  - Well validated, automated update and roll-back strategy between versions of K8s
  - Allows for ultimate configurability for enabling cloud best practices for security are followed
- Disadvantages:
  - You’re on your own for issue diagnosis
  - Higher management overhead

Kubernetes is Good for Ygrene

Ygrene is Great for the Planet

Come work with us!

ygrene.com/careers

ygrene.tech