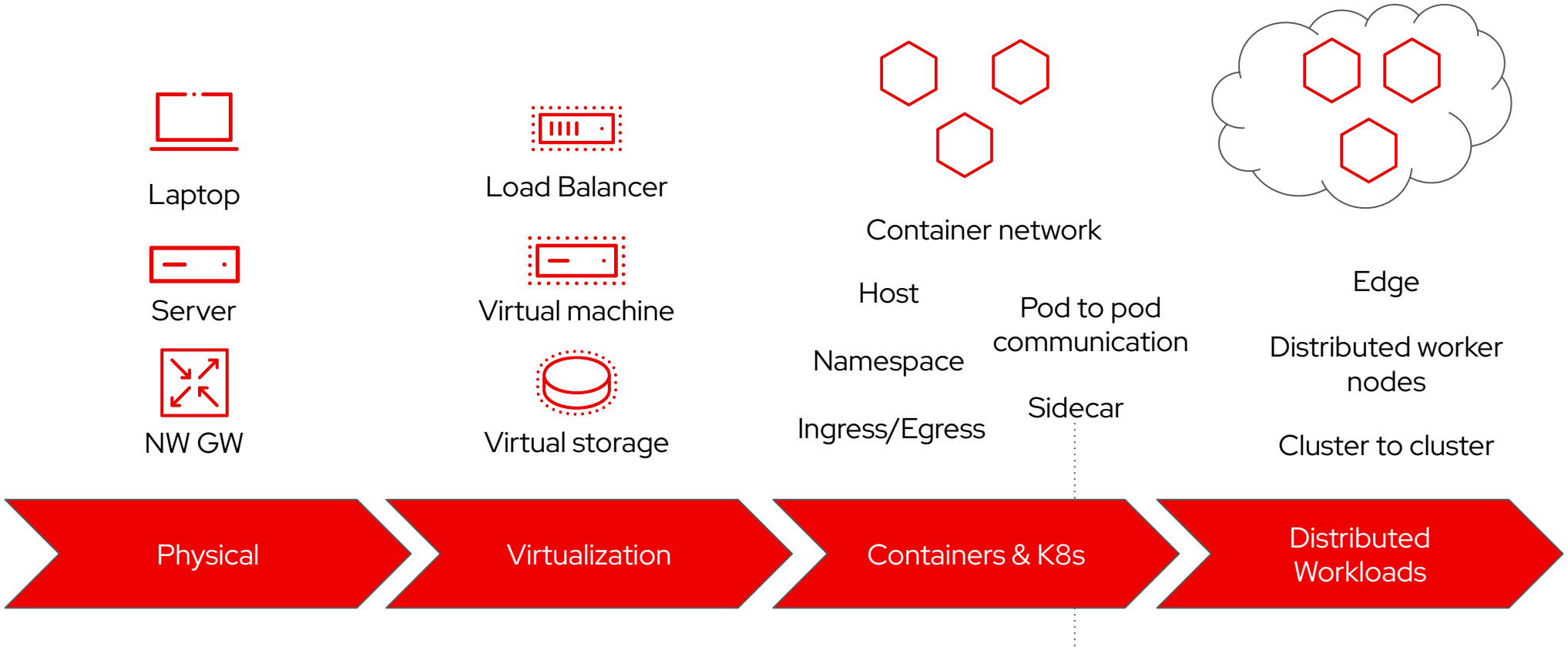
Zero Trust Security for Cloud Native Environments

Maria Simon Marcos
Senior Product Manager
Hybrid Platforms, Red Hat

Our reality

The concept of 'perimeter' is changing



Red Hat

A layered approach

Security posture

Traditional approach

- Based on a trusted zone/untrusted zones
- Reactive Security:
  - Deny list policy
  - Demilitarized zones
  - Antivirus
  - Intrusion systems

Zero trust approach

- Based on no implicit trust. Authentication and authorization required between all parties. Identity, Integrity, Isolation
- Proactive Security:
  - Live segmentation
  - Supply chain security
  - Policy as code / Security automation
  - Continuous compliance enforcement

Red Hat

Defense in depth

Building better **walls**

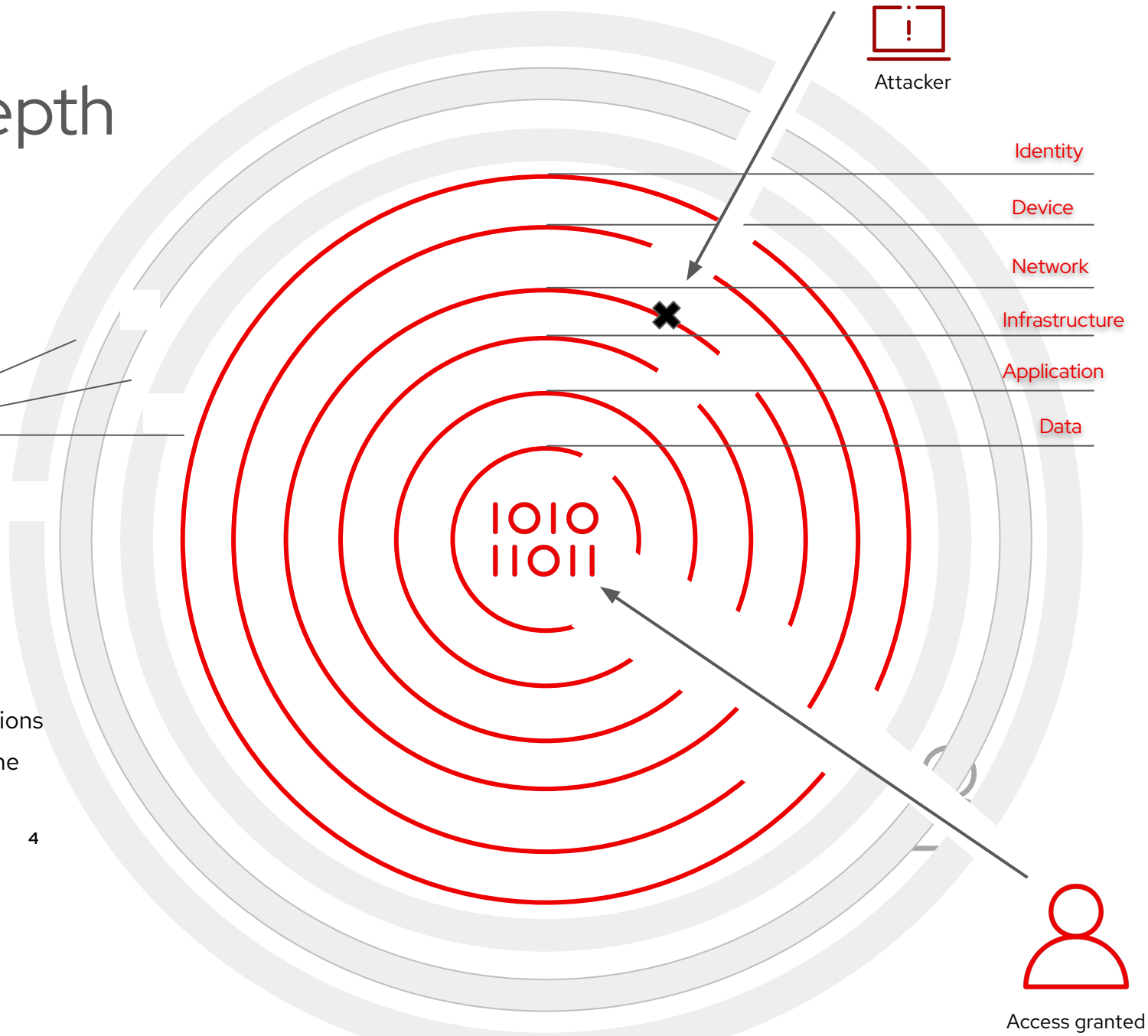
Zero trust

Additional **gates**

Walls
=
security controls

Unknown risks:

- Mitigated via overlapping, diverse controls
- Complementary implementations
- Continuous assessments of the status of the controls
- Adversarial testing to find uncovered holes



Red Hat

What's is the foundation of the Zero Trust model?

7 pillars based on your assets

Automated policy decisions

Automation & Orchestration

Visibility & Analytics

Constant monitoring & logging combined with threat intelligence and analytics

Data

Data categorization & encryption

Microsegmentation and isolation

Network

Security wrapped around each workload, supply chain security

Application & Workload

Device

Identification & authorization of devices

Identity

Authentication & access control policies

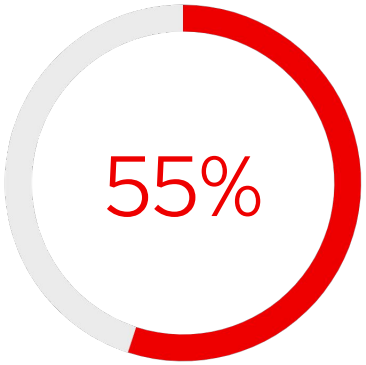
Red Hat

State of Kubernetes security

Security is a business-level discussion

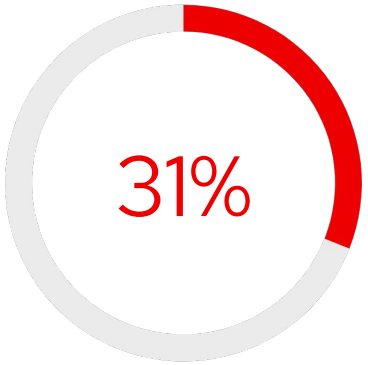
Security remains one of the top concerns with container and Kubernetes strategies.

Left unaddressed, Kubernetes security challenges can inhibit business performance



Security concerns slow innovation

55% delayed or slowed down application deployment into production due to Kubernetes security concerns



Security issues hinder growth

31% suffered customer or revenue loss due to a Kubernetes security or compliance issue

Source:
State of Kubernetes Security 2022

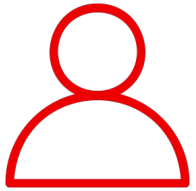
Red Hat

Cloud Native Zero Trust Security

Let's look at the 7 pillars

Red Hat

Identity



Identity and Access Management

- Use identity provider with 2FA
- Different access for humans & machines
- Manage cluster access and application access separately
- Least privilege role based access controls

Workload identity = Service Accounts

- Separate service accounts per application
- Audit service accounts in your cluster

Automation & Orchestration
Visibility & Analytics
Data
Network
Application & Workload
Device
Identity

Red Hat

Device (Host)

Secure Boot

- Provides guarantee that a trusted, unmodified kernel is loaded
- Advanced Instrusion Detection Engine (AIDE) to monitor file integrity and detect system intrusions

Volume encryption

- Addresses disk/image theft, including etcd datastore
- NBDE : Network-Bound Disk Encryption via Clevis and Tang
- TPM: Trusted Platform Module Encryption

Hardening

- Use a container optimized operating system to minimize attack surface
- Harden the host environment

Automation & Orchestration

Visibility & Analytics

Data

Network

Application & Workload

Device

Identity

Red Hat

Application & Workload

Supply Chain Security

- Use a trusted container base image
- Secure Supply Chain Validated Pattern Architecture
  - Image signing (Sigstore - Cosign)
  - Scanning of deployment configurations and Helm charts (KubeLinter)
  - Deployment signing
  - Tekton Chains helps automate the process

Limiting Linux Capabilities

- OCP SCCs: Limit container runtime privileges in the pod spec using security context constraints.
- Run your cluster on a container optimized host OS such as RHCOS that comes with SElinux.
  - For other Linux distros, use App Armor.
- (Advanced) Define seccomp profile to filter system call availability within a container

Automation & Orchestration

Visibility & Analytics

Data

Network

Application & Workload

Device

Identity

Red Hat

Network

Microsegment and Isolate

- Multi-tenancy 'resource' isolation using different namespaces
- Restrict pod to pod in cluster traffic with Kubernetes Network policies
- Use Ingress controller to expose services outside the cluster
- Use Egress controls to restrict traffic leaving your cluster:
  - Egress IP
  - Egress Firewall
- Different TLS security profiles can be chosen per component in OCP

Service Mesh

- Easier service-to-service (east-west traffic) communication permission by service name
- Service identities are authenticated using mTLS
- Sidecar proxies attached to the service collect metrics and log all communication and access request between services

Automation & Orchestration

Visibility & Analytics

Data

Network

Application & Workload

Device

Identity

Red Hat

Data

Secret management

- Decide where are you going to store your secrets:
  - etcd
  - External secrets store
  - KMS (Key Management System)
- Integrations:
  - External Secrets Operator provides for external management systems eg. Azure Key Vault, AWS Secrets Manager...
- Decide if you want to encrypt:
  - Etcd
  - Disk

Encryption in transit

- Pod to pod communication
  - Service-to-service communication with mTLS -> Service Mesh
- Data from/to outside the cluster
  - Ingress controller and API Gateways in Service Mesh help expose services via SSL/TLS

Automation & Orchestration

Visibility & Analytics

Data

Network

Application & Workload

Device

Identity

Red Hat

Visibility & Analytics

Logging

- Different logs available:
  - Host OS
  - API server
  - Cluster component
  - Application
- Select the logging verbose most applicable for your use case
- Forward logging to a external server or your SIEM of choice

Monitoring

- Scan often your workloads images for vulnerabilities
- Monitor overly permissive access in your workloads
- Health monitoring of your cluster
- Malicious activity - Runtime behavior analysis
- Identify misconfigurations
- Automated compliance

Visibility

- Use single pane of glass and multi-cluster visibility to facilitate understanding of metrics and trends over time:
  - Security policy violations
  - Compliance
  - Vulnerability management
  - Risk profiles
  - Network communication

Automation & Orchestration

Visibility & Analytics

Data

Network

Application & Workload

Device

Identity

Red Hat

Automation & Orchestration

Policy based Governance

- Policy-based application pipelines with built-in security gates
- Use policy as code and GitOps Integration
- Integration of admission controllers into the policy framework. Eg. OPA Gatekeeper

Automation

- Automated enforcement of the desired state of your clusters
- Perform remediation actions by leveraging Ansible Automation Platform (Automated governance)
- Automate compliance checks and remediation against security standards such: CIS benchmarks and NIST 800-53

Automation & Orchestration

Visibility & Analytics

Data

Network

Application & Workload

Device

Identity

Red Hat

OpenShift delivers continuous security

	Control	Protect	Detect & Respond
ACM	Application Lifecycle and Locality	Fleet Management	Fleet Observability & Alerts
ACS	Vulnerability analysis	Policy admission controller	Runtime behavioral analysis
ACS	App config analysis	Compliance assessments	Auto-suggest network policies
ACS	APIs for CI/CD integrations	Risk profiling	Threat detection / incident response
OCP	Trusted content	Kubernetes platform lifecycle	Container isolation
OCP	Container registry	Identity and access management	Network isolation
OCP	Build management	Protect platform data	Protect application access and data
OCP	CI/CD pipeline	Deployment policies	Observability
	BUILD	DEPLOY	RUN

DevSecOps

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make
Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat