StackRox

Mitigating Kubernetes Attacks

Wei Lien Dang – Co-founder and Chief Strategy Officer

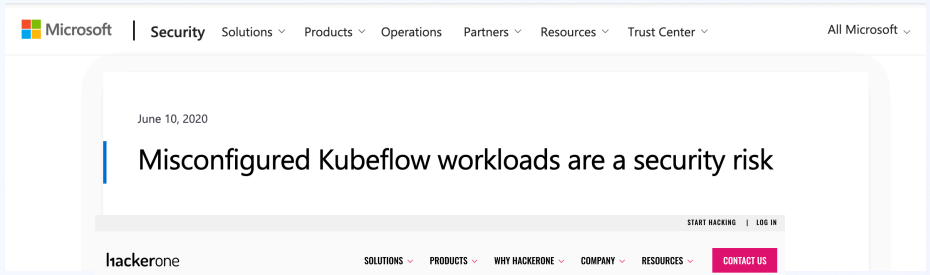
Michelle McLean – VP of Marketing

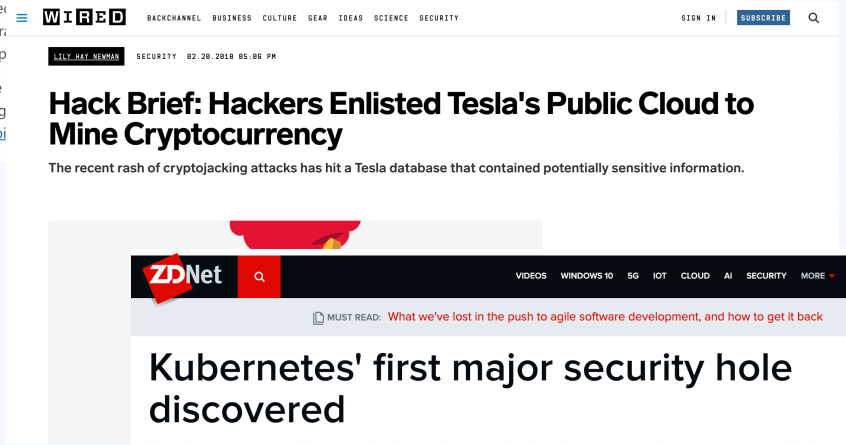
September 23, 2020

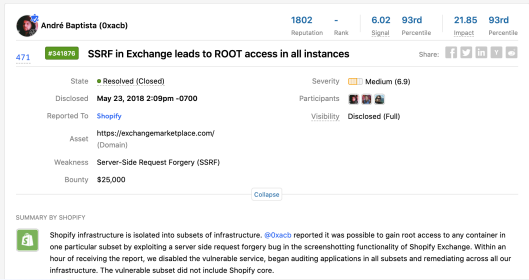
Agenda

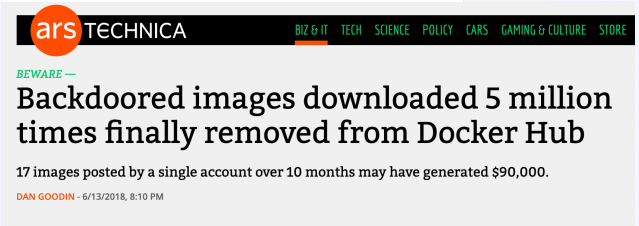
- Kubernetes and container attacks: real-world examples
- Kubernetes threats: what's different?
- Kubernetes attack matrix: overview
- Adversarial tactics and techniques: main themes
- Key takeaways and recommendations

Kubernetes attacks in the wild









StackRox research

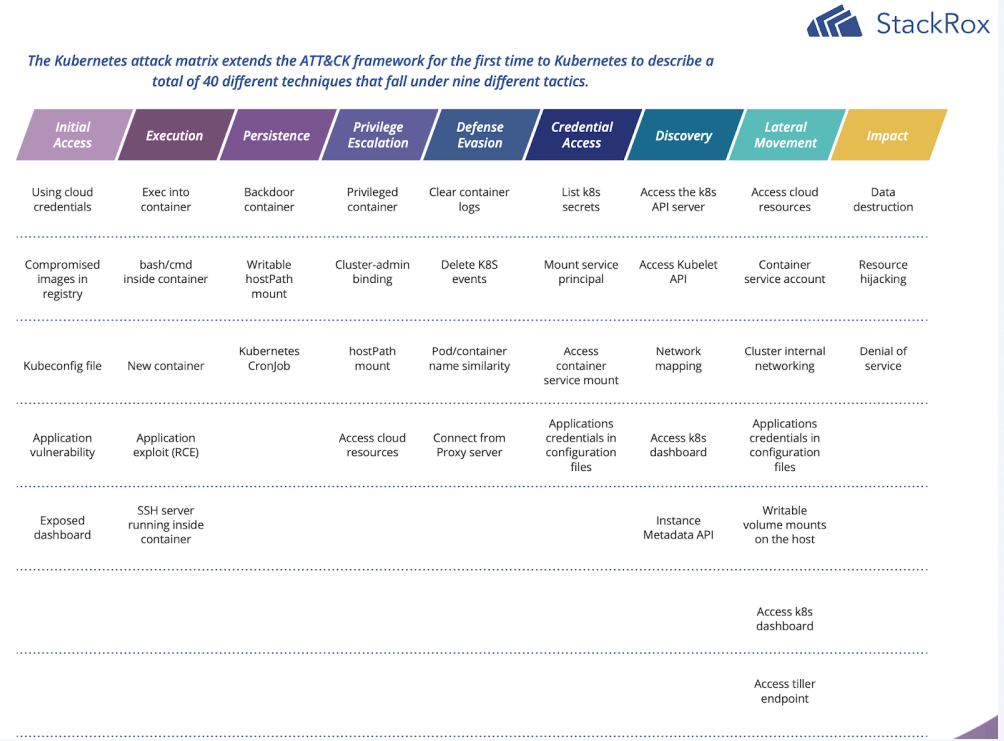
- Honeypot setup
  - Large GKE clusters running hundreds of containerized apps exposed to the Internet for 5 months
  - Used popular images with known vulnerabilities and deployed with weak configurations
- Threats look similar to those that affect non-containerized applications
- Observed attacker actions
  - Injection attempts to download a file into /tmp/
  - Attempted downloads using wget
  - Intrusion attempts frequently occurring on well-known web ports
  - Attempted commands to gain additional targeting data or download binaries

What's different about Kubernetes threats

- New attack surface
  - Kubernetes control plane
  - Cluster worker nodes
- Application components are highly distributed, dynamic, and ephemeral
- Increased operational complexity
- Broader impact due to orchestration and automation

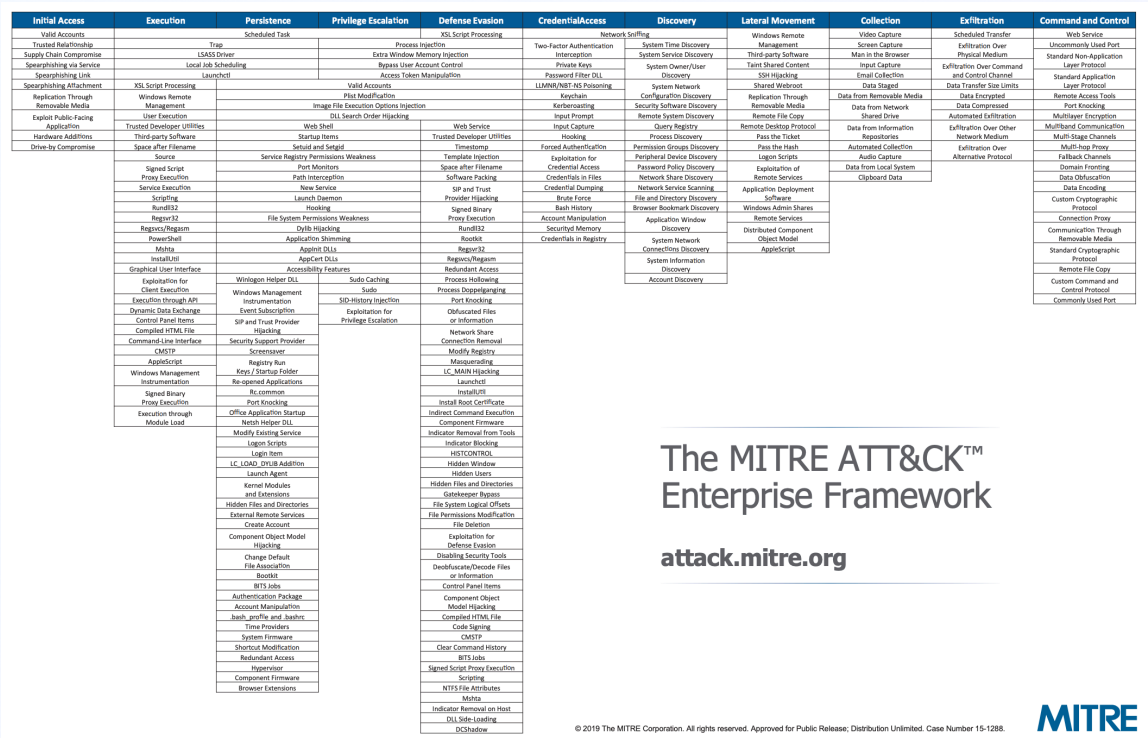
Kubernetes attack matrix: an overview

- Published by Microsoft Azure
- Part of ecosystem's continued focus on security
  - Kubernetes security audit
  - SIGs
  - NIST SP 800-190
  - CIS Kubernetes Benchmark
- Based on MITRE ATT&CK® framework

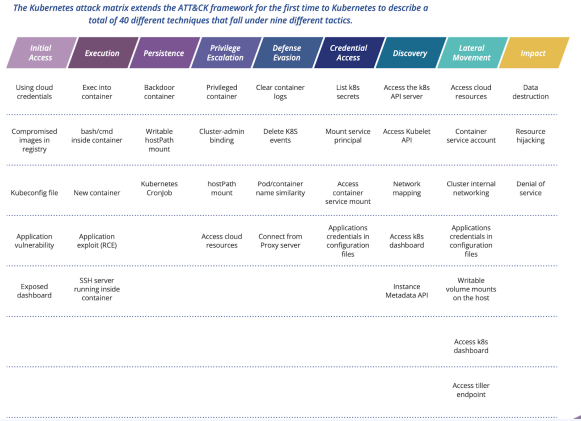


MITRE ATT&CK® framework

- Knowledge base of adversarial tactics and techniques
- Use to categorize attack vectors and gauge their level of risk
- Based on real-world observations



Tactics and techniques of the attack matrix



- 9 tactics, 40 techniques
- Tactics: the “why” behind a particular technique
- Techniques: specific offensive actions - the “how” for a given objective
- Some techniques can be classified under multiple tactics
- A technique may warrant multiple, different mitigations

Examples: 1.4 Application vulnerability, 2.4. Application exploit

- Scan container images for vulnerabilities
- Use admission control to prevent containers with high-severity vulnerabilities from launching
- Configure Network Policies to limit external access to pods
- Restrict service account permissions using Kubernetes RBAC
- Do not allow pods to run as root
- Set up filesystem as read-only
- Minimize container access to underlying host

Your best protection: apply native Kubernetes controls

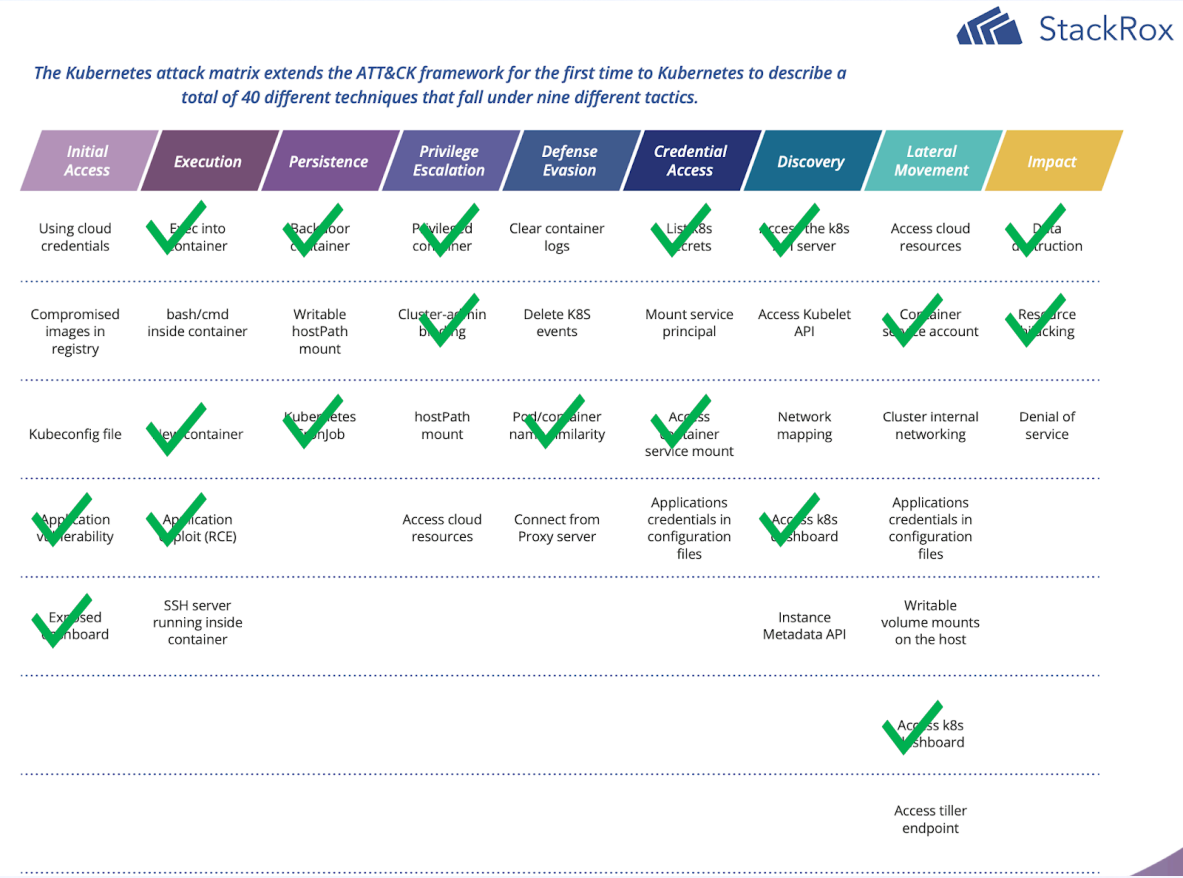
StackRox

The Kubernetes attack matrix extends the ATT&CK framework for the first time to Kubernetes to describe a total of 40 different techniques that fall under nine different tactics.

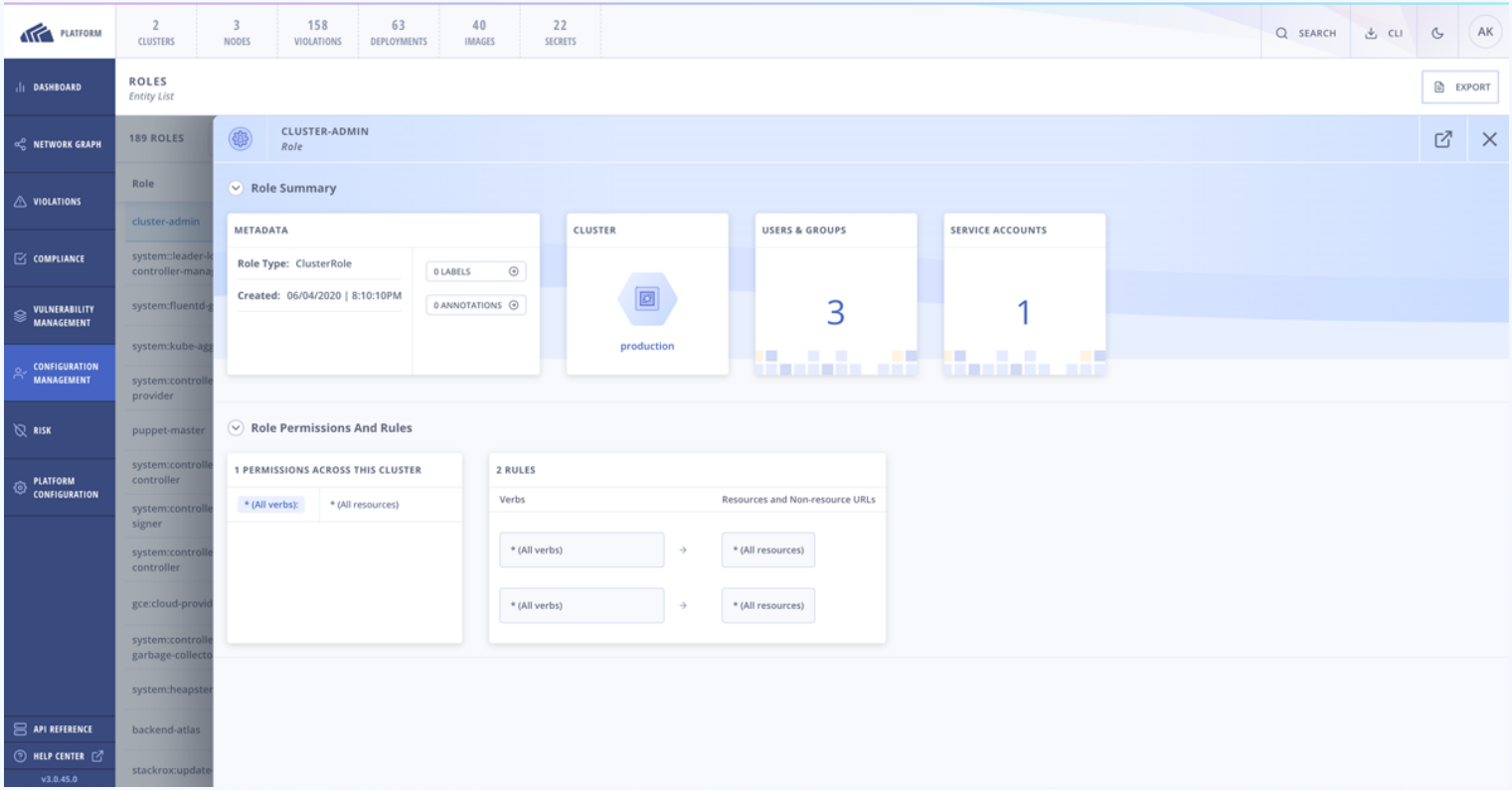
Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
Using cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List k8s secrets	Access the k8s API server	Access cloud resources	Data destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete k8S events	Mount service principal	Access Kubelet API	Container service account	Resource hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod/container name similarity	Access container service mount	Network mapping	Cluster-internal networking	Denial of service
Application vulnerability	Application exploit (RCE)		Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access k8s dashboard	Applications credentials in configuration files	
Exposed dashboard	SSH server running inside container					Instance Metadata API	Writable volume mounts on the host	
							Access k8s dashboard	
							Access tiller endpoint	

Step 1: Configure Kubernetes RBAC

- Limit who has the cluster-admin role in your organization
- Adopt a least-privilege model for service accounts and their role bindings
- Avoid complexity in role aggregation or overlap in role definitions

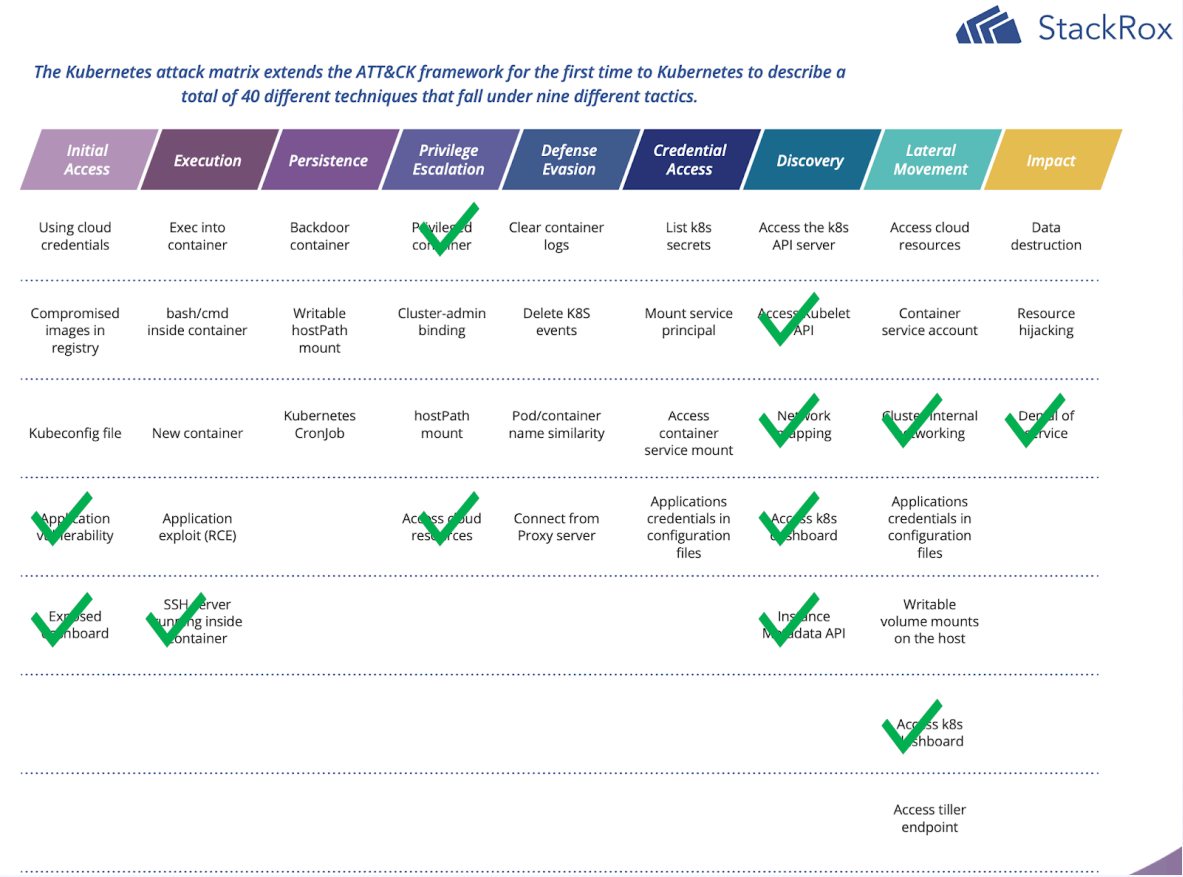


Ensure you monitor your RBAC settings

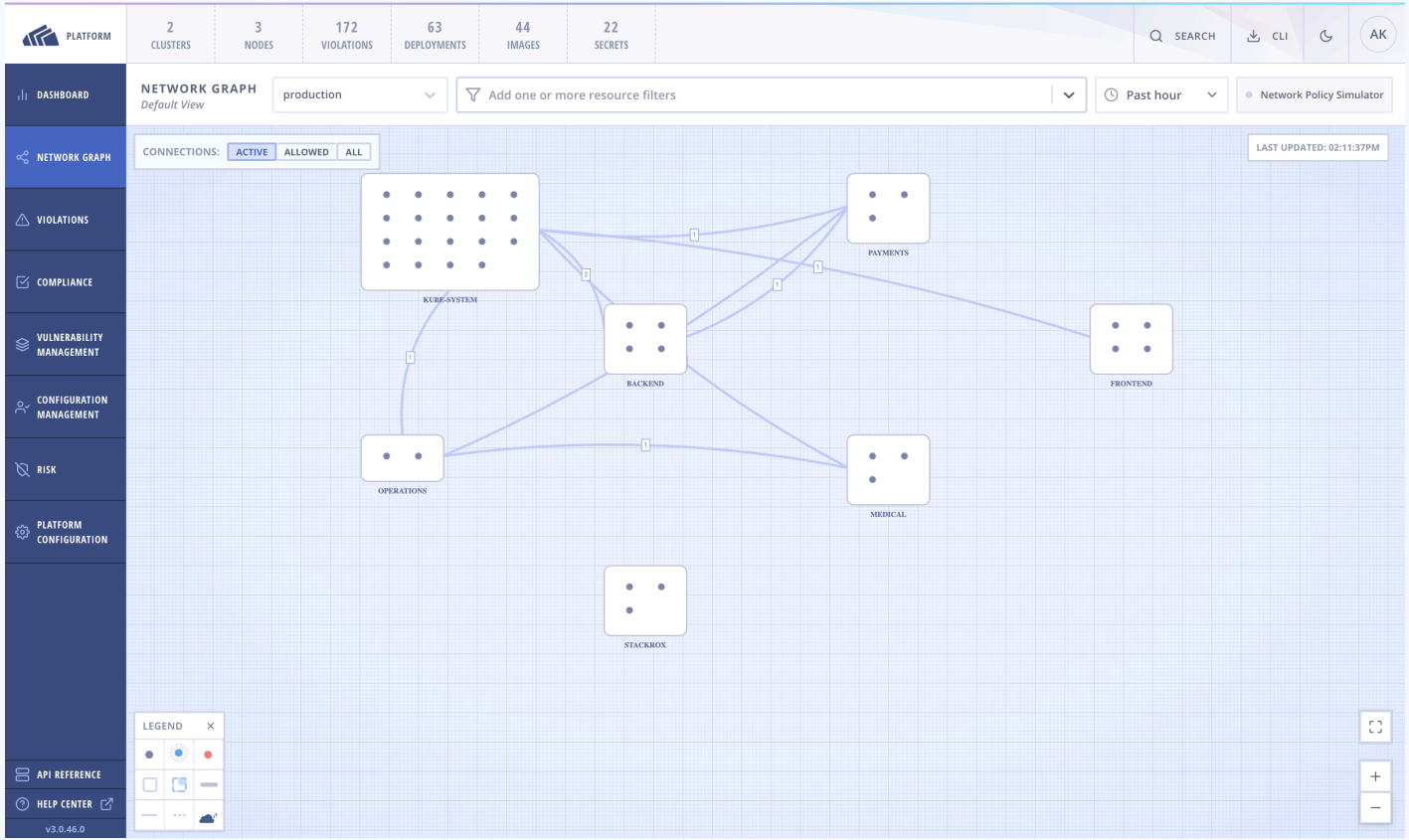


Step 2: Configure Network Policies

- Use a CNI that implements the NetworkPolicy API and create policies that restrict pod traffic
- Start by applying a default-deny-all network policy
- Explicitly allow necessary Internet access and pod-to-pod communication

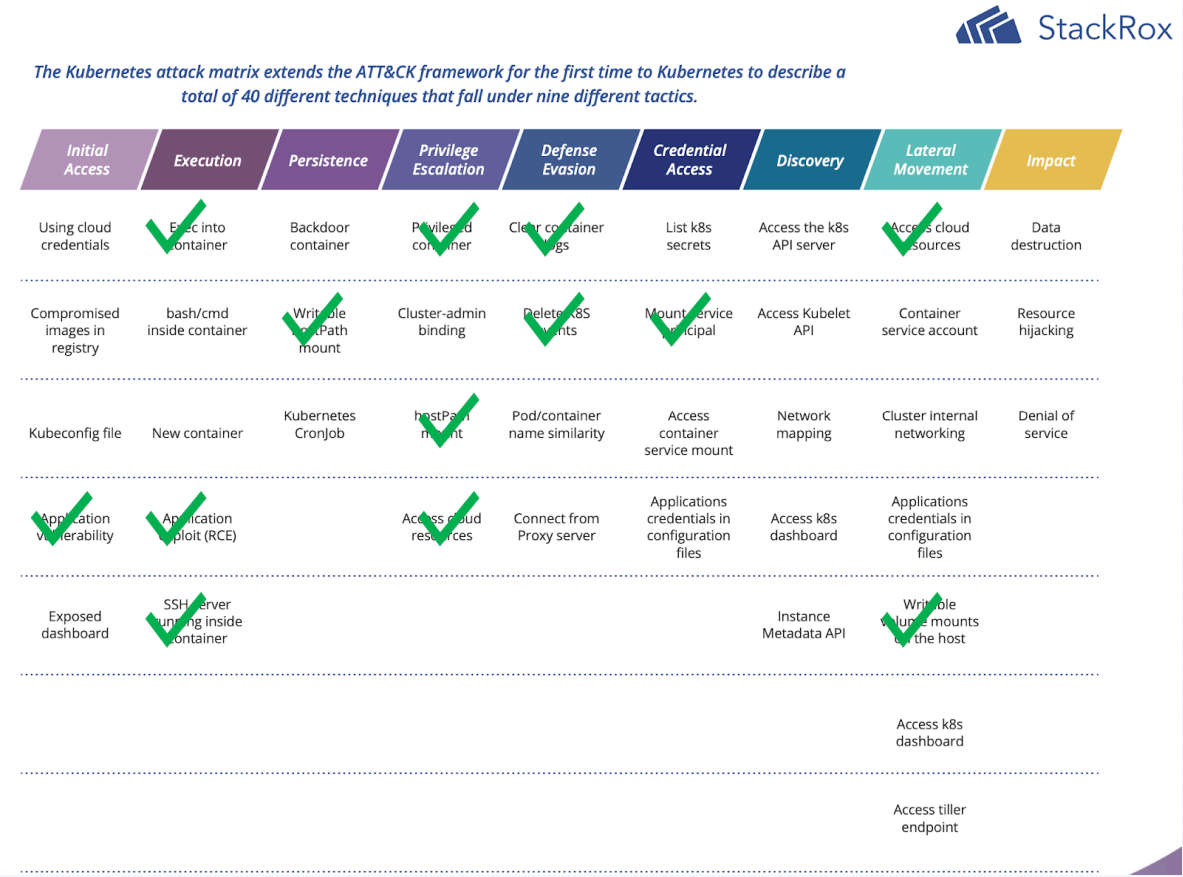


Look for ways to automate Network Policy management

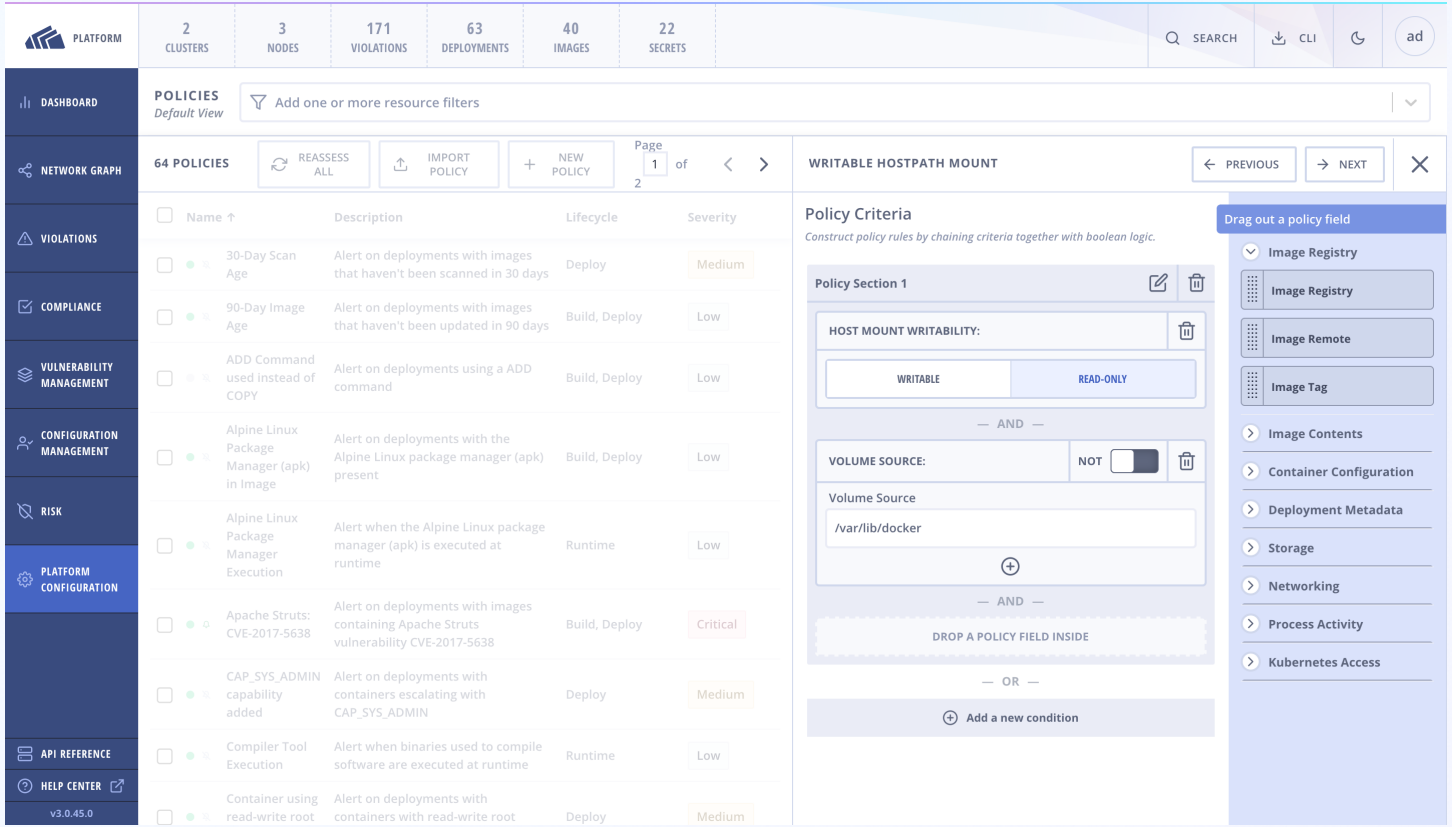


Step 3: Harden pod configurations

- Configure security contexts for pods and/or containers
- Enforce policies on pod specifications
- Authorize policies by granting access to the pod's service account



Detect insecure pod configurations



Key takeaways and recommendations

- Use the Kubernetes attack matrix as your basis for systematically and comprehensively securing your containerized applications

- Applying a few native Kubernetes security features will mitigate most attacks:
  - Kubernetes RBAC
  - Kubernetes Network Policies
  - Restricted pod configurations

- You still need runtime monitoring!

Q&A