cast

// K8s security

Security Hardening Checklist For Kubernetes

Looking to secure your Kubernetes cluster? Here's a list of the most important security boxes you should check to sleep well at night knowing that your deployment is fully secure.

1 Check exposed services for forbidden types

We're talking about things like NodePort or LoadBalancer. It's best to expose K8s services through ClusterIP – that way, you prevent the discovery of cluster infrastructure components.

2 Make sure images aren't using the "latest" or "blank" tag

Some attacks are based on downloading public images with the "latest" tag. Avoid using "latest" or no tag when defining images to be used by a pod. You'll also make tracking the container version easier.

3 Check that your secrets are encrypted

Secrets aren't encrypted at rest by default in base Kubernetes implementations. If anyone intercepts your key-value store, they'll get access to everything in your cluster - including cluster secrets in plain text! Encrypt to secure your cluster against data-at-rest exfiltration.

4 Protect secrets in environment variables

It's smart to have your objects use a secret in an environment variable as other parts of your system can access these variables. Use secrets as files or take advantage of a secretKeyRef to minimize potential attacks.

5 Secure access to etcd

Access to etcd is like root permission access. It's the most important piece to secure within the K8s control plane. Make sure that communication with etcd is encrypted and clients use certificate-based authentication. To limit the attack surface, let only the API server access etcd.

6 Use read-only root FS

Using a read-only file system prevents malicious binaries from writing to a system or attackers from system takeover. You can ensure that containers use just the read-only file system by setting readOnlyRootFilesystem to true in the Pod securityContext definition.

7 Keep sensitive host mounts to a minimum

Are deployments with sensitive host system directories like /, /boot, /dev, /etc, /lib, /proc, /sys, /usr, mounted in your containers? This is tricky. It might lead to changes in sensitive files within these directories and have security implications. It's better to avoid mounting these directories to a container unless it's absolutely necessary.

8 Check if the SSH port is not exposed

Does your K8s deployment expose port 22? This port is commonly reserved for SSH access – it's often an attack target for attackers. Avoid exposing it and sleep better at night.

9 Enhance the control plane's security

Even if your cloud provider manages the control plane components, you remain responsible for securing your nodes, containers, and pods.

By default, the Kubernetes API server uses a public IP address. You can protect it by using authorized networks and private clusters, which enable you to assign a private IP address.

You can also enhance your control plane's security by doing a regular credential rotation. When you initiate the process, the TLS certificates and cluster certificate authority are rotated automatically.

10 Use strong authentication and authorization

Integrating Kubernetes with a third-party authentication provider is smart, as it brings you extra security features like multi-factor authentication. Avoid managing users at the API server level to make access to the control plane more secure.

Use the Identity and Access Management (IAM) solution from the cloud service provider if you're running a managed Kubernetes service like EKS, GKE, AKS, or similar. Go for OpenID Connect (OIDC) alongside an SSO provider you got used to if CSP IAM isn't an option.

As the Kubernetes ecosystem evolves, so do its security concerns. Keeping up with changes is time-consuming, and once vulnerabilities pile up, engineers are forced to prioritize many items at once.

CAST AI's Security Report Best Practices feature checks clusters against industry best practices, Kubernetes recommendations, and CIS Kubernetes benchmarks – and then prioritizes them automatically to set you on the right track from the start.