Kubernetes Threat Model

Privilege Escalation

Privilege escalation occurs when an attacker gains elevated permissions, allowing them to perform actions that should be restricted. In Kubernetes, preventing privilege escalation is crucial to maintaining the security and integrity of the cluster and its applications.

Kubernetes provides several mechanisms to prevent privilege escalation, including security contexts, Role-Based Access Control (RBAC), and PodSecurityPolicies (PSPs). Properly configuring these controls helps ensure that users and applications operate with the minimum necessary permissions, reducing the risk of privilege escalation.

RealLife Example:

Imagine a lowly servant (regular user) in the kingdom who usurps power and seizes control. In Kubernetes, privilege escalation occurs when an attacker with low privileges manages to gain higher privileges, potentially compromising the entire cluster.

Key Concepts

1. **Security Contexts**
   - Define security settings for pods and containers, such as user privileges, capabilities, and file system permissions.
   - Use security contexts to enforce least privilege and prevent privilege escalation.
2. **Role-Based Access Control (RBAC)**
   - RBAC controls who can access and manipulate Kubernetes resources.
   - Use RBAC to define fine-grained permissions and enforce the principle of least privilege.
3. **PodSecurityPolicies (PSPs)**
   - PSPs are cluster-level resources that control security-sensitive aspects of pod specifications.
   - Use PSPs to enforce security policies and prevent privilege escalation.

4. **Admission Controllers**
    - Admission controllers govern and enforce policies on how the cluster is used.
    - Use controllers like PodSecurityPolicy, SecurityContextDeny, and ImagePolicyWebhook to prevent privilege escalation.

Security Best Practices

1. **Use Security Contexts**
    - Define security contexts to limit container privileges and capabilities.
    - Run containers as non-root users and enforce read-only file systems.
2. **Implement RBAC for Access Control**
    - Define roles and role bindings to restrict access to sensitive resources.
    - Use the principle of least privilege to grant only necessary permissions.
3. **Enforce PodSecurityPolicies (PSPs)**
    - Create and enforce PSPs to restrict pod capabilities and configurations.
    - Ensure pods run with the minimum required privileges.
4. **Enable Admission Controllers**
    - Configure admission controllers to validate and enforce security policies.
    - Use controllers to deny insecure configurations and enforce best practices.
5. **Monitor and Audit Access**
    - Enable audit logging to track access and modifications to critical resources.
    - Regularly review logs to detect unauthorized access and potential privilege escalation attempts.