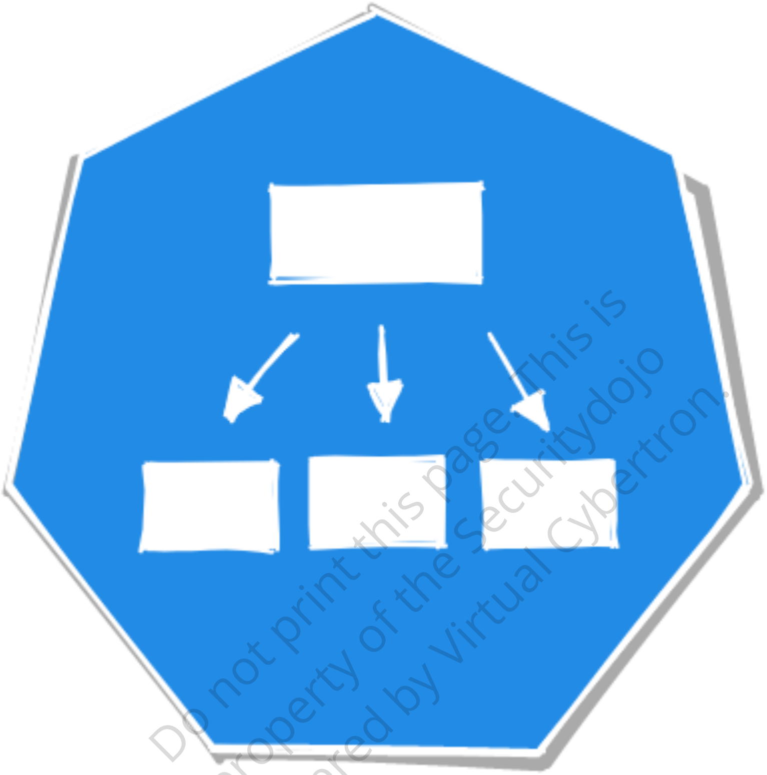
Attacking Role Based Access Controls



Role-Based Access Control (RBAC) is the primary authorization mechanism in Kubernetes, responsible for managing permissions over resources within the cluster. RBAC combines actions (verbs) such as get, create, and delete with resources like pods, services, and nodes to define access control.

The example provided (vulapp-clusterrole-full) demonstrates a misconfigured RBAC, where full access is granted to all resources and actions. This configuration is potentially dangerous and not recommended, as it can lead to security vulnerabilities.

Prevention

- Regularly audit RBAC configurations, especially those related to third-party components, to ensure they adhere to the principle of least privilege.
- Minimize direct cluster access for end users and avoid using Service Account Tokens outside the cluster to limit potential attack vectors.
- Use RoleBindings to restrict permissions to specific namespaces, rather than applying cluster-wide RBAC policies, for better access control.
- Follow the official Kubernetes RBAC Good Practices documentation to maintain a secure and well-configured RBAC environment.

Reference:

- OWASP