Kubernetes RBAC Security Best Practices

# 1. Role and RoleBinding Security Best Practices

Role and RoleBinding are used to grant access to resources within a specific namespace. Below are the best practices to ensure security when assigning Roles:  
  
- Principle of Least Privilege: Only assign permissions that are absolutely required for the user or service account.  
- Namespace Isolation: Prefer Role and RoleBinding when working within a namespace. Avoid ClusterRole if not needed.  
- Service Account Scoping: Bind Roles to service accounts rather than users when automating application permissions.  
- Audit Bindings: Regularly audit RoleBindings to ensure that there are no excessive or outdated permissions.  
- Avoid Wildcards: Minimize use of `\*` in verbs, API groups, or resources to reduce scope.

# 2. ClusterRole and ClusterRoleBinding Security Best Practices

ClusterRoles are cluster-scoped and allow access to resources across all namespaces. Best practices include:  
  
- Use Caution with ClusterRoleBinding: Bind users to ClusterRoles only when cluster-wide access is essential.  
- Avoid system:masters Group: Do not assign users or service accounts to `system:masters` unless they are cluster administrators.  
- Create Custom ClusterRoles: Define custom ClusterRoles for specific use-cases instead of relying on default ones.  
- Periodic Reviews: Review ClusterRoleBindings to ensure they are still valid and not over-provisioned.  
- Logging and Auditing: Enable audit logging in the cluster to monitor use of ClusterRoles and ClusterRoleBindings.

# 3. General Security Recommendations

- Avoid using default service accounts for workloads.  
- Monitor RBAC changes with GitOps or audit tools.  
- Document RBAC permissions as part of application deployment processes.  
- Use tools like `kubectl-who-can` or `rakkess` to validate access rights.