```
[INFO] 3 Worker Node Security Configuration
[INFO] 3.1 Worker Node Configuration Files
[PASS] 3.1.1 Ensure that the kubeconfig file permissions are set to 644 or more
restrictive (Manual)
[PASS] 3.1.2 Ensure that the kubelet kubeconfig file ownership is set to root:root
(Manual)
[PASS] 3.1.3 Ensure that the kubelet configuration file has permissions set to 644
or more restrictive (Manual)
[PASS] 3.1.4 Ensure that the kubelet configuration file ownership is set to
root:root (Manual)
[INFO] 3.2 Kubelet
[PASS] 3.2.1 Ensure that the --anonymous-auth argument is set to false (Automated)
[PASS] 3.2.2 Ensure that the --authorization-mode argument is not set to
AlwaysAllow (Automated)
[PASS] 3.2.3 Ensure that the --client-ca-file argument is set as appropriate
(Manual)
[PASS] 3.2.4 Ensure that the --read-only-port argument is set to 0 (Manual)
[PASS] 3.2.5 Ensure that the --streaming-connection-idle-timeout argument is not
set to 0 (Manual)
[PASS] 3.2.6 Ensure that the --protect-kernel-defaults argument is set to true
(Automated)
[PASS] 3.2.7 Ensure that the --make-iptables-util-chains argument is set to true
(Automated)
[PASS] 3.2.8 Ensure that the --hostname-override argument is not set (Manual)
[WARN] 3.2.9 Ensure that the --eventRecordQPS argument is set to 0 or a level which
ensures appropriate event capture (Automated)
[PASS] 3.2.10 Ensure that the --rotate-certificates argument is not set to false
(Manual)
[PASS] 3.2.11 Ensure that the RotateKubeletServerCertificate argument is set to
true (Manual)

== Remediations node ==
3.2.9 If using a Kubelet config file, edit the file to set eventRecordQPS: to an
appropriate level.
If using command line arguments, edit the kubelet service file
/etc/systemd/system/kubelet.service on each worker node and
set the below parameter in KUBELET_SYSTEM_PODS_ARGS variable.
Based on your system, restart the kubelet service. For example:
systemctl daemon-reload
systemctl restart kubelet.service


== Summary node ==
14 checks PASS
0 checks FAIL
1 checks WARN
0 checks INFO

[INFO] 4 Policies
[INFO] 4.1 RBAC and Service Accounts
```

[WARN] 4.1.1 Ensure that the cluster-admin role is only used where required (Manual)
[WARN] 4.1.2 Minimize access to secrets (Manual)
[WARN] 4.1.3 Minimize wildcard use in Roles and ClusterRoles (Manual)
[WARN] 4.1.4 Minimize access to create pods (Manual)
[WARN] 4.1.5 Ensure that default service accounts are not actively used. (Manual)
[WARN] 4.1.6 Ensure that Service Account Tokens are only mounted where necessary (Manual)
[INFO] 4.2 Pod Security Policies
[WARN] 4.2.1 Minimize the admission of privileged containers (Automated)
[WARN] 4.2.2 Minimize the admission of containers wishing to share the host process ID namespace (Automated)
[WARN] 4.2.3 Minimize the admission of containers wishing to share the host IPC namespace (Automated)
[WARN] 4.2.4 Minimize the admission of containers wishing to share the host network namespace (Automated)
[WARN] 4.2.5 Minimize the admission of containers with allowPrivilegeEscalation (Automated)
[WARN] 4.2.6 Minimize the admission of root containers (Automated)
[WARN] 4.2.7 Minimize the admission of containers with the NET_RAW capability (Automated)
[WARN] 4.2.8 Minimize the admission of containers with added capabilities (Automated)
[WARN] 4.2.9 Minimize the admission of containers with capabilities assigned (Manual)
[INFO] 4.3 CNI Plugin
[WARN] 4.3.1 Ensure that the latest CNI version is used (Manual)
[WARN] 4.3.2 Ensure that all Namespaces have Network Policies defined (Automated)
[INFO] 4.4 Secrets Management
[WARN] 4.4.1 Prefer using secrets as files over secrets as environment variables (Manual)
[WARN] 4.4.2 Consider external secret storage (Manual)
[INFO] 4.5 Extensible Admission Control
[WARN] 4.5.1 Configure Image Provenance using ImagePolicyWebhook admission controller (Manual)
[INFO] 4.6 General Policies
[WARN] 4.6.1 Create administrative boundaries between resources using namespaces (Manual)
[WARN] 4.6.2 Apply Security Context to Your Pods and Containers (Manual)
[WARN] 4.6.3 The default namespace should not be used (Automated)

== Remediations policies ==
4.1.1 Identify all clusterrolebindings to the cluster-admin role. Check if they are used and
if they need this role or if they could use a role with fewer privileges.
Where possible, first bind users to a lower privileged role and then remove the clusterrolebinding to the cluster-admin role :
kubectl delete clusterrolebinding [name]

4.1.2 Where possible, remove get, list and watch access to secret objects in the

cluster.

4.1.3 Where possible replace any use of wildcards in clusterroles and roles with specific
objects or actions.

4.1.4 Where possible, remove create access to pod objects in the cluster.

4.1.5 Create explicit service accounts wherever a Kubernetes workload requires specific access
to the Kubernetes API server.
Modify the configuration of each default service account to include this value
automountServiceAccountToken: false

4.1.6 Modify the definition of pods and service accounts which do not need to mount service
account tokens to disable it.

4.2.1 Create a PSP as described in the Kubernetes documentation, ensuring that
the .spec.privileged field is omitted or set to false.

4.2.2 Create a PSP as described in the Kubernetes documentation, ensuring that the
.spec.hostPID field is omitted or set to false.

4.2.3 Create a PSP as described in the Kubernetes documentation, ensuring that the
.spec.hostIPC field is omitted or set to false.

4.2.4 Create a PSP as described in the Kubernetes documentation, ensuring that the
.spec.hostNetwork field is omitted or set to false.

4.2.5 Create a PSP as described in the Kubernetes documentation, ensuring that the
.spec.allowPrivilegeEscalation field is omitted or set to false.

4.2.6 Create a PSP as described in the Kubernetes documentation, ensuring that the
.spec.runAsUser.rule is set to either MustRunAsNonRoot or MustRunAs with the range of
UIDs not including 0.

4.2.7 Create a PSP as described in the Kubernetes documentation, ensuring that the
.spec.requiredDropCapabilities is set to include either NET_RAW or ALL.

4.2.8 Ensure that allowedCapabilities is not present in PSPs for the cluster unless
it is set to an empty array.

4.2.9 Review the use of capabilities in applications running on your cluster. Where a namespace
contains applications which do not require any Linux capabities to operate consider adding
a PSP which forbids the admission of containers which do not drop all capabilities.

4.3.1 Review the documentation of AWS CNI plugin, and ensure latest CNI version is used.

4.3.2 Follow the documentation and create NetworkPolicy objects as you need them.

4.4.1 If possible, rewrite application code to read secrets from mounted secret files, rather than
from environment variables.

4.4.2 Refer to the secrets management options offered by your cloud provider or a third-party
secrets management solution.

4.5.1 Follow the Kubernetes documentation and setup image provenance.

4.6.1 Follow the documentation and create namespaces for objects in your deployment as you need
them.

4.6.2 Follow the Kubernetes documentation and apply security contexts to your pods. For a
suggested list of security contexts, you may refer to the CIS Security Benchmark for Docker
Containers.

4.6.3 Ensure that namespaces are created to allow for appropriate segregation of Kubernetes
resources and that all new resources are created in a specific namespace.

== Summary policies ==
0 checks PASS
0 checks FAIL
23 checks WARN
0 checks INFO

[INFO] 5 Managed Services
[INFO] 5.1 Image Registry and Image Scanning
[WARN] 5.1.1 Ensure Image Vulnerability Scanning using Amazon ECR image scanning or a third-party provider (Manual)
[WARN] 5.1.2 Minimize user access to Amazon ECR (Manual)
[WARN] 5.1.3 Minimize cluster access to read-only for Amazon ECR (Manual)
[WARN] 5.1.4 Minimize Container Registries to only those approved (Manual)
[INFO] 5.2 Identity and Access Management (IAM)
[WARN] 5.2.1 Prefer using dedicated Amazon EKS Service Accounts (Manual)
[INFO] 5.3 AWS Key Management Service (KMS)
[WARN] 5.3.1 Ensure Kubernetes Secrets are encrypted using Customer Master Keys (CMKs) managed in AWS KMS (Manual)
[INFO] 5.4 Cluster Networking
[WARN] 5.4.1 Restrict Access to the Control Plane Endpoint (Manual)

```
[WARN] 5.4.2 Ensure clusters are created with Private Endpoint Enabled and Public
Access Disabled (Manual)
[WARN] 5.4.3 Ensure clusters are created with Private Nodes (Manual)
[WARN] 5.4.4 Ensure Network Policy is Enabled and set as appropriate (Manual)
[WARN] 5.4.5 Encrypt traffic to HTTPS load balancers with TLS certificates (Manual)
[INFO] 5.5 Authentication and Authorization
[WARN] 5.5.1 Manage Kubernetes RBAC users with AWS IAM Authenticator for Kubernetes
(Manual)
[INFO] 5.6 Other Cluster Configurations
[WARN] 5.6.1 Consider Fargate for running untrusted workloads (Manual)

== Remediations managedservices ==
5.1.1 No remediation
5.1.2 No remediation
5.1.3 No remediation
5.1.4 No remediation
5.2.1 No remediation
5.3.1 No remediation
5.4.1 No remediation
5.4.2 No remediation
5.4.3 No remediation
5.4.4 No remediation
5.4.5 No remediation
5.5.1 No remediation
5.6.1 No remediation

== Summary managedservices ==
0 checks PASS
0 checks FAIL
13 checks WARN
0 checks INFO

== Summary total ==
14 checks PASS
0 checks FAIL
37 checks WARN
0 checks INFO
```