[INFO] 4 Worker Node Security Configuration [INFO] 4.1 Worker Node Configuration Files [FAIL] 4.1.1 Ensure that the kubelet service file permissions are set to 600 or more restrictive (Automated) [PASS] 4.1.2 Ensure that the kubelet service file ownership is set to root:root (Automated) [PASS] 4.1.3 If proxy kubeconfig file exists ensure permissions are set to 600 or more restrictive (Manual) [PASS] 4.1.4 If proxy kubeconfig file exists ensure ownership is set to root:root (Manual) [PASS] 4.1.5 Ensure that the –kubeconfig kubelet.conf file permissions are set to 600 or more restrictive (Automated) [PASS] 4.1.6 Ensure that the –kubeconfig kubelet.conf file ownership is set to root:root (Automated) [WARN] 4.1.7 Ensure that the certificate authorities file permissions are set to 600 or more restrictive (Manual) [WARN] 4.1.8 Ensure that the client certificate authorities file ownership is set to root:root (Manual) [WARN] 4.1.9 If the kubelet config.yaml configuration file is being used validate permissions set to 600 or more restrictive (Manual) [PASS] 4.1.10 If the kubelet config.yaml configuration file is being used validate file ownership is set to root:root (Manual) [INFO] 4.2 Kubelet [FAIL] 4.2.1 Ensure that the –anonymous-auth argument is set to false (Automated) [FAIL] 4.2.2 Ensure that the –authorization-mode argument is not set to AlwaysAllow (Automated) [FAIL] 4.2.3 Ensure that the –client-ca-file argument is set as appropriate (Automated) [PASS] 4.2.4 Verify that the –read-only-port argument is set to 0 (Manual) [PASS] 4.2.5 Ensure that the –streaming-connection-idle-timeout argument is not set to 0 (Manual) [FAIL] 4.2.6 Ensure that the –protect-kernel-defaults argument is set to true (Automated) [PASS] 4.2.7 Ensure that the –make-iptables-util-chains argument is set to true (Automated) [WARN] 4.2.8 Ensure that the –hostname-override argument is not set (Manual) [PASS] 4.2.9 Ensure that the eventRecordQPS argument is set to a level which ensures appropriate event capture (Manual) [WARN] 4.2.10 Ensure that the –tls-cert-file and –tls-private-key-file arguments are set as appropriate (Manual) [PASS] 4.2.11 Ensure that the –rotate-certificates argument is not set to false (Automated) [PASS] 4.2.12 Verify that the RotateKubeletServerCertificate argument is set to true (Manual) [WARN] 4.2.13 Ensure that the Kubelet only makes use of Strong Cryptographic Ciphers (Manual)

== Remediations node == 4.1.1 Run the below command (based on the file location on your system) on the each worker node. For example, chmod 600 /etc/systemd/system/kubelet.service

4.1.7 Run the following command to modify the file permissions of the –client-ca-file chmod 600

4.1.8 Run the following command to modify the ownership of the –client-ca-file. chown root:root

4.1.9 Run the following command (using the config file location identified in the Audit step) chmod 600 /etc/systemd/system/kubelet.service

4.2.1 If using a Kubelet config file, edit the file to set `authentication: anonymous: enabled` to `false`. If using executable 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. `--anonymous-auth=false` Based on your system, restart the kubelet service. For example, systemctl daemon-reload systemctl restart kubelet.service

4.2.2 If using a Kubelet config file, edit the file to set `authorization.mode` to Webhook. If using executable arguments, edit the kubelet service file /etc/systemd/system/kubelet.service on each worker node and set the below parameter in KUBELET_AUTHZ_ARGS variable. –authorization-mode=Webhook Based on your system, restart the kubelet service. For example, systemctl daemon-reload systemctl restart kubelet.service

4.2.3 If using a Kubelet config file, edit the file to set `authentication.x509.clientCAFile` to the location of the client CA file. 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_AUTHZ_ARGS variable. –client-ca-file= Based on your system, restart the kubelet service. For example, systemctl daemon-reload systemctl restart kubelet.service

4.2.6 If using a Kubelet config file, edit the file to set `protectKernelDefaults` to `true`. 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. –protect-kernel-defaults=true Based on your system, restart the kubelet service. For example: systemctl daemon-reload systemctl restart kubelet.service

4.2.8 Edit the kubelet service file /etc/systemd/system/kubelet.service on each worker node and remove the –hostname-override argument from the KUBELET_SYSTEM_PODS_ARGS variable. Based on your system, restart the kubelet service. For example, systemctl daemon-reload systemctl restart kubelet.service

4.2.10 If using a Kubelet config file, edit the file to set `tlsCertFile` to the location of the certificate file to use to identify this Kubelet, and `tlsPrivateKeyFile` to the location of the corresponding private key file. If using command line arguments, edit the kubelet service file /etc/systemd/system/kubelet.service on each worker node and set the below parameters in KUBELET_CERTIFICATE_ARGS variable. –tls-cert-file= –tls-private-key-file= Based on your system, restart the kubelet service. For example, systemctl daemon-reload systemctl restart kubelet.service

4.2.13 If using a Kubelet config file, edit the file to set `TLSCipherSuites` to TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_S or to a subset of these values. If using executable arguments, edit the kubelet service file /etc/systemd/system/kubelet.service on each worker node and set the –tls-cipher-suites parameter as follows, or to a subset of these values. –tls-cipher-suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_G Based on your system, restart the kubelet service. For example: systemctl daemon-reload systemctl restart kubelet.service

== Summary node == 12 checks PASS 5 checks FAIL 6 checks WARN 0 checks INFO

[INFO] 5 Kubernetes Policies [INFO] 5.1 RBAC and Service Accounts [WARN] 5.1.1 Ensure that the cluster-admin role is only used where required (Manual) [WARN] 5.1.2 Minimize access to secrets (Manual) [WARN] 5.1.3 Minimize wildcard use in Roles and ClusterRoles (Manual) [WARN] 5.1.4 Minimize access to create pods (Manual) [WARN] 5.1.5 Ensure that default service accounts are not actively used. (Manual) [WARN] 5.1.6 Ensure that Service Account Tokens are only mounted where necessary (Manual) [WARN] 5.1.7 Avoid use of system:masters group (Manual) [WARN] 5.1.8 Limit use of the Bind, Impersonate and Escalate permissions in the Kubernetes cluster (Manual) [INFO] 5.2 Pod Security Standards [WARN] 5.2.1 Ensure that the cluster has at least one active policy control mechanism in place (Manual) [WARN] 5.2.2 Minimize the admission of privileged containers (Manual) [WARN] 5.2.3 Minimize the admission of containers wishing to share the host process ID namespace (Automated) [WARN] 5.2.4 Minimize the admission of containers wishing to share the host IPC namespace (Automated) [WARN] 5.2.5 Minimize the admission of containers wishing to share the host network namespace (Automated) [WARN] 5.2.6 Minimize the admission of containers with allowPrivilegeEscalation (Automated) [WARN] 5.2.7 Minimize the admission of root containers (Automated) [WARN] 5.2.8 Minimize the admission of containers with the NET_RAW capability (Automated) [WARN] 5.2.9 Minimize the admission of containers with added capabilities (Automated) [WARN] 5.2.10 Minimize the admission of containers with capabilities assigned (Manual) [WARN] 5.2.11 Minimize the admission of Windows HostProcess containers (Manual) [WARN] 5.2.12 Minimize the admission of HostPath volumes (Manual) [WARN] 5.2.13 Minimize the admission of containers which use HostPorts (Manual) [INFO] 5.3 Network Policies and CNI [WARN] 5.3.1 Ensure that the CNI in use supports NetworkPolicies (Manual) [WARN] 5.3.2 Ensure that all Namespaces have NetworkPolicies defined (Manual) [INFO] 5.4 Secrets Management [WARN] 5.4.1 Prefer using Secrets as files over Secrets as environment variables (Manual) [WARN] 5.4.2 Consider external secret storage (Manual) [INFO] 5.5 Extensible Admission Control [WARN] 5.5.1 Configure Image Provenance using ImagePolicyWebhook admission controller (Manual) [INFO] 5.7 General Policies [WARN] 5.7.1 Create administrative boundaries between resources using namespaces (Manual) [WARN] 5.7.2 Ensure that the seccomp profile is set to docker/default in your Pod definitions (Manual) [WARN] 5.7.3 Apply SecurityContext to your Pods and Containers (Manual) [WARN] 5.7.4 The default namespace should not be used (Manual)

== Remediations policies == 5.1.1 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]

5.1.2 Where possible, remove get, list and watch access to Secret objects in the

cluster.

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

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

5.1.5 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

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

5.1.7 Remove the system:masters group from all users in the cluster.

5.1.8 Where possible, remove the impersonate, bind and escalate rights from subjects.

5.2.1 Ensure that either Pod Security Admission or an external policy control system is in place for every namespace which contains user workloads.

5.2.2 Add policies to each namespace in the cluster which has user workloads to restrict the admission of privileged containers.

5.2.3 Add policies to each namespace in the cluster which has user workloads to restrict the admission of `hostPID` containers.

5.2.4 Add policies to each namespace in the cluster which has user workloads to restrict the admission of `hostIPC` containers.

5.2.5 Add policies to each namespace in the cluster which has user workloads to restrict the admission of `hostNetwork` containers.

5.2.6 Add policies to each namespace in the cluster which has user workloads to restrict the admission of containers with `.spec.allowPrivilegeEscalation` set to `true`.

5.2.7 Create a policy for each namespace in the cluster, ensuring that either `MustRunAsNonRoot` or `MustRunAs` with the range of UIDs not including 0, is set.

5.2.8 Add policies to each namespace in the cluster which has user workloads to restrict the admission of containers with the `NET_RAW` capability.

5.2.9 Ensure that `allowedCapabilities` is not present in policies for the cluster unless it is set to an empty array.

5.2.10 Review the use of capabilites in applications running on your cluster. Where a namespace contains applicaions 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.

5.2.11 Add policies to each namespace in the cluster which has user workloads to restrict the admission of containers that have `.securityContext.windowsOptions.hostProcess` set to `true`.

5.2.12 Add policies to each namespace in the cluster which has user workloads to restrict the admission of containers with `hostPath` volumes.

5.2.13 Add policies to each namespace in the cluster which has user workloads to restrict the admission of containers which use `hostPort` sections.

5.3.1 If the CNI plugin in use does not support network policies, consideration should be given to making use of a different plugin, or finding an alternate mechanism for restricting traffic in the Kubernetes cluster.

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

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

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

5.5.1 Follow the Kubernetes documentation and setup image provenance.

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

5.7.2 Use `securityContext` to enable the docker/default seccomp profile in your pod definitions. An example is as below: securityContext: seccompProfile: type: RuntimeDefault

5.7.3 Follow the Kubernetes documentation and apply SecurityContexts to your Pods. For a suggested list of SecurityContexts, you may refer to the CIS Security Benchmark for Docker Containers.

5.7.4 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 30 checks WARN 0 checks INFO

== Summary total == 12 checks PASS 5 checks FAIL 36 checks WARN 0 checks INFO