Kubernetes CIS Benchmark Report

docker-desktop

Date: 12-06-2025 15:26:52

Summary total

Status	Count
PASS	60
FAIL	17
WARN	53
INFO	0

1 Control Plane Security Configuration

1.1 Control Plane Node Configuration Files

Test ID: 1.1.1 Status: PASS

1.1.1 Ensure that the API server pod specification file permissions are set to 600 or more restrictive (Automated)

Test ID: 1.1.2 Status: PASS

1.1.2 Ensure that the API server pod specification file ownership is set to root:root (Automated)

Test ID: 1.1.3 Status: PASS

1.1.3 Ensure that the controller manager pod specification file permissions are set to 600 or more restrictive (Automated)

Test ID: 1.1.4 Status: PASS

1.1.4 Ensure that the controller manager pod specification file ownership is set to root:root (Automated)

Test ID: 1.1.5 Status: PASS

1.1.5 Ensure that the scheduler pod specification file permissions are set to 600 or more restrictive (Automated)

Test ID: 1.1.6 Status: PASS

1.1.6 Ensure that the scheduler pod specification file ownership is set to root:root (Automated)

Test ID: 1.1.7 Status: PASS

1.1.7 Ensure that the etcd pod specification file permissions are set to 600 or more restrictive (Automated)

Test ID: 1.1.8 Status: PASS

1.1.8 Ensure that the etcd pod specification file ownership is set to root:root (Automated)

Test ID: 1.1.9 Status: WARN

1.1.9 Ensure that the Container Network Interface file permissions are set to 600 or more restrictive (Manual)

Test ID: 1.1.10 Status: PASS

1.1.10 Ensure that the Container Network Interface file ownership is set to root:root (Manual)

Test ID: 1.1.11 Status: PASS

1.1.11 Ensure that the etcd data directory permissions are set to 700 or more restrictive (Automated)

Test ID: 1.1.12 Status: FAIL

1.1.12 Ensure that the etcd data directory ownership is set to etcd:etcd (Automated)

Test ID: 1.1.13 Status: PASS

1.1.13 Ensure that the default administrative credential file permissions are set to 600 (Automated)

Test ID: 1.1.14 Status: PASS

1.1.14 Ensure that the default administrative credential file ownership is set to root:root (Automated)

Test ID: 1.1.15 Status: PASS

1.1.15 Ensure that the scheduler.conf file permissions are set to 600 or more restrictive (Automated)

Test ID: 1.1.16 Status: PASS

1.1.16 Ensure that the scheduler.conf file ownership is set to root:root (Automated)

Test ID: 1.1.17 Status: PASS

1.1.17 Ensure that the controller-manager.conf file permissions are set to 600 or more restrictive (Automated)

Test ID: 1.1.18 Status: PASS

1.1.18 Ensure that the controller-manager.conf file ownership is set to root:root (Automated)

Test ID: 1.1.19 Status: PASS

1.1.19 Ensure that the Kubernetes PKI directory and file ownership is set to root:root (Automated)

Test ID: 1.1.20 Status: WARN

1.1.20 Ensure that the Kubernetes PKI certificate file permissions are set to 600 or more restrictive (Manual)

Test ID: 1.1.21 Status: WARN

1.1.21 Ensure that the Kubernetes PKI key file permissions are set to 600 (Manual)

1.2 API Server

Test ID: 1.2.1 Status: WARN

1.2.1 Ensure that the --anonymous-auth argument is set to false (Manual)

Test ID: 1.2.2 Status: PASS

1.2.2 Ensure that the --token-auth-file parameter is not set (Automated)

Test ID: 1.2.3 Status: WARN

1.2.3 Ensure that the --DenyServiceExternalIPs is set (Manual)

Test ID: 1.2.4 Status: PASS

1.2.4 Ensure that the --kubelet-client-certificate and --kubelet-client-key arguments are set as appropriate (Automated)

Test ID: 1.2.5 Status: FAIL

1.2.5 Ensure that the --kubelet-certificate-authority argument is set as appropriate (Automated)

Test ID: 1.2.6 Status: PASS

1.2.6 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Automated)

Test ID: 1.2.7 Status: PASS

1.2.7 Ensure that the --authorization-mode argument includes Node (Automated)

Test ID: 1.2.8 Status: PASS

1.2.8 Ensure that the --authorization-mode argument includes RBAC (Automated)

Test ID: 1.2.9 Status: WARN

1.2.9 Ensure that the admission control plugin EventRateLimit is set (Manual)

Test ID: 1.2.10 Status: PASS

1.2.10 Ensure that the admission control plugin AlwaysAdmit is not set (Automated)

Test ID: 1.2.11 Status: WARN

1.2.11 Ensure that the admission control plugin AlwaysPullImages is set (Manual)

Test ID: 1.2.12 Status: PASS

1.2.12 Ensure that the admission control plugin ServiceAccount is set (Automated)

Test ID: 1.2.13 Status: PASS

1.2.13 Ensure that the admission control plugin NamespaceLifecycle is set (Automated)

Test ID: 1.2.14 Status: PASS

1.2.14 Ensure that the admission control plugin NodeRestriction is set (Automated)

Test ID: 1.2.15 Status: FAIL

1.2.15 Ensure that the --profiling argument is set to false (Automated)

Test ID: 1.2.16 Status: FAIL

1.2.16 Ensure that the --audit-log-path argument is set (Automated)

Test ID: 1.2.17 Status: FAIL

1.2.17 Ensure that the --audit-log-maxage argument is set to 30 or as appropriate (Automated)

Test ID: 1.2.18 Status: FAIL

1.2.18 Ensure that the --audit-log-maxbackup argument is set to 10 or as appropriate (Automated)

Test ID: 1.2.19 Status: FAIL

1.2.19 Ensure that the --audit-log-maxsize argument is set to 100 or as appropriate (Automated)

Test ID: 1.2.20 Status: WARN

1.2.20 Ensure that the --request-timeout argument is set as appropriate (Manual)

Test ID: 1.2.21 Status: PASS

1.2.21 Ensure that the --service-account-lookup argument is set to true (Automated)

Test ID: 1.2.22 Status: PASS

1.2.22 Ensure that the --service-account-key-file argument is set as appropriate (Automated)

Test ID: 1.2.23 Status: PASS

1.2.23 Ensure that the --etcd-certfile and --etcd-keyfile arguments are set as appropriate (Automated)

Test ID: 1.2.24 Status: PASS

1.2.24 Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate (Automated)

Test ID: 1.2.25 Status: PASS

1.2.25 Ensure that the --client-ca-file argument is set as appropriate (Automated)

Test ID: 1.2.26 Status: PASS

1.2.26 Ensure that the --etcd-cafile argument is set as appropriate (Automated)

Test ID: 1.2.27 Status: WARN

1.2.27 Ensure that the --encryption-provider-config argument is set as appropriate (Manual)

Test ID: 1.2.28 Status: WARN

1.2.28 Ensure that encryption providers are appropriately configured (Manual)

Test ID: 1.2.29 Status: WARN

1.2.29 Ensure that the API Server only makes use of Strong Cryptographic Ciphers (Manual)

1.3 Controller Manager

Test ID: 1.3.1 Status: WARN

1.3.1 Ensure that the --terminated-pod-gc-threshold argument is set as appropriate (Manual)

Test ID: 1.3.2 Status: FAIL

1.3.2 Ensure that the --profiling argument is set to false (Automated)

Test ID: 1.3.3 Status: PASS

1.3.3 Ensure that the --use-service-account-credentials argument is set to true (Automated)

Test ID: 1.3.4 Status: PASS

1.3.4 Ensure that the --service-account-private-key-file argument is set as appropriate (Automated)

Test ID: 1.3.5 Status: PASS

1.3.5 Ensure that the --root-ca-file argument is set as appropriate (Automated)

Test ID: 1.3.6 Status: PASS

1.3.6 Ensure that the RotateKubeletServerCertificate argument is set to true (Automated)

Test ID: 1.3.7 Status: PASS

1.3.7 Ensure that the --bind-address argument is set to 127.0.0.1 (Automated)

1.4 Scheduler

Test ID: 1.4.1 Status: FAIL

1.4.1 Ensure that the --profiling argument is set to false (Automated)

Test ID: 1.4.2 Status: PASS

1.4.2 Ensure that the --bind-address argument is set to 127.0.0.1 (Automated)

== Remediations master ==
1.1.9 Run the below command (based on the file location on your system) on the control plane node.
For example, chmod 600 <path/to/cni/files>

1.1.12 On the etcd server node, get the etcd data directory, passed as an argument --data-dir,
from the command 'ps -ef | grep etcd'.
Run the below command (based on the etcd data directory found above).
For example, chown etcd:etcd /var/lib/etcd

1.1.20 Run the below command (based on the file location on your system) on the control plane node.
For example,
chmod -R 600 /etc/kubernetes/pki/*.crt

1.1.21 Run the below command (based on the file location on your system) on the control plane node.
For example,
chmod -R 600 /etc/kubernetes/pki/*.key

1.2.1 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and set the below parameter.
--anonymous-auth=false

1.2.3 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and add the `DenyServiceExternalIPs` plugin
to the enabled admission plugins, as such --enable-admission-plugin=DenyServiceExternalIPs.

1.2.5 Follow the Kubernetes documentation and setup the TLS connection between
the apiserver and kubelets. Then, edit the API server pod specification file
/etc/kubernetes/manifests/kube-apiserver.yaml on the control plane node and set the
--kubelet-certificate-authority parameter to the path to the cert file for the certificate authority.
--kubelet-certificate-authority=<ca-string>

1.2.9 Follow the Kubernetes documentation and set the desired limits in a configuration file.
Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
and set the below parameters.
--enable-admission-plugins=...,EventRateLimit,...
--admission-control-config-file=<path/to/configuration/file>

1.2.11 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and set the --enable-admission-plugins parameter to include
AlwaysPullImages.
--enable-admission-plugins=...,AlwaysPullImages,...

1.2.15 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and set the below parameter.
--profiling=false

1.2.16 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and set the --audit-log-path parameter to a suitable path and
file where you would like audit logs to be written, for example,
--audit-log-path=/var/log/apiserver/audit.log

1.2.17 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and set the --audit-log-maxage parameter to 30
or as an appropriate number of days, for example,
--audit-log-maxage=30

1.2.18 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and set the --audit-log-maxbackup parameter to 10 or to an appropriate
value. For example,
--audit-log-maxbackup=10

1.2.19 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and set the --audit-log-maxsize parameter to an appropriate size in MB.

For example, to set it as 100 MB, --audit-log-maxsize=100

1.2.20 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
and set the below parameter as appropriate and if needed.
For example, --request-timeout=300s

1.2.27 Follow the Kubernetes documentation and configure a EncryptionConfig file.
Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and set the --encryption-provider-config parameter to the path of that file.
For example, --encryption-provider-config=</path/to/EncryptionConfig/File>

1.2.28 Follow the Kubernetes documentation and configure a EncryptionConfig file.
In this file, choose aescbc, kms or secretbox as the encryption provider.

1.2.29 Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml
on the control plane node and set the below parameter.
--tls-cipher-suites=TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256

1.3.1 Edit the Controller Manager pod specification file /etc/kubernetes/manifests/kube-controller-manager.yaml
on the control plane node and set the --terminated-pod-gc-threshold to an appropriate threshold,
for example, --terminated-pod-gc-threshold=10

1.3.2 Edit the Controller Manager pod specification file /etc/kubernetes/manifests/kube-controller-manager.yaml
on the control plane node and set the below parameter.
--profiling=false

1.4.1 Edit the Scheduler pod specification file /etc/kubernetes/manifests/kube-scheduler.yaml file
on the control plane node and set the below parameter.
--profiling=false

Summary master

Status	Count
PASS	38
FAIL	9
WARN	12
INFO	0

2 Etcd Node Configuration

Test ID: 2.1 Status: PASS

2.1 Ensure that the --cert-file and --key-file arguments are set as appropriate (Automated)

Test ID: 2.2 Status: PASS

2.2 Ensure that the --client-cert-auth argument is set to true (Automated)

Test ID: 2.3 Status: PASS

2.3 Ensure that the --auto-tls argument is not set to true (Automated)

Test ID: 2.4 Status: PASS

2.4 Ensure that the --peer-cert-file and --peer-key-file arguments are set as appropriate (Automated)

Test ID: 2.5 Status: PASS

2.5 Ensure that the --peer-client-cert-auth argument is set to true (Automated)

Test ID: 2.6 Status: PASS

2.6 Ensure that the --peer-auto-tls argument is not set to true (Automated)

Test ID: 2.7 Status: PASS

2.7 Ensure that a unique Certificate Authority is used for etcd (Manual)

Summary etcd

Status	Count
PASS	7
FAIL	0
WARN	0
INFO	0

3 Control Plane Configuration

3.1 Authentication and Authorization

Test ID: 3.1.1 Status: WARN

3.1.1 Client certificate authentication should not be used for users (Manual)

Test ID: 3.1.2 Status: WARN

3.1.2 Service account token authentication should not be used for users (Manual)

Test ID: 3.1.3 Status: WARN

3.1.3 Bootstrap token authentication should not be used for users (Manual)

3.2 Logging

Test ID: 3.2.1 Status: WARN

3.2.1 Ensure that a minimal audit policy is created (Manual)

Test ID: 3.2.2 Status: WARN

3.2.2 Ensure that the audit policy covers key security concerns (Manual)

== Remediations controlplane ==
3.1.1 Alternative mechanisms provided by Kubernetes such as the use of OIDC should be implemented in place of client certificates.

3.1.2 Alternative mechanisms provided by Kubernetes such as the use of OIDC should be implemented in place of service account tokens.

3.1.3 Alternative mechanisms provided by Kubernetes such as the use of OIDC should be implemented in place of bootstrap tokens.

3.2.1 Create an audit policy file for your cluster.

3.2.2 Review the audit policy provided for the cluster and ensure that it covers at least the following areas,
- Access to Secrets managed by the cluster. Care should be taken to only log Metadata for requests to Secrets, ConfigMaps, and TokenReviews, in order to avoid risk of logging sensitive data.
- Modification of Pod and Deployment objects.
- Use of `pods/exec`, `pods/portforward`, `pods/proxy` and `services/proxy`.
For most requests, minimally logging at the Metadata level is recommended (the most basic level of logging).

Summary controlplane

Status	Count
PASS	0
FAIL	0
WARN	5
INFO	0

4 Worker Node Security Configuration

4.1 Worker Node Configuration Files

Test ID: 4.1.1 Status: FAIL

4.1.1 Ensure that the kubelet service file permissions are set to 600 or more restrictive (Automated)

Test ID: 4.1.2 Status: PASS

4.1.2 Ensure that the kubelet service file ownership is set to root:root (Automated)

Test ID: 4.1.3 Status: WARN

4.1.3 If proxy kubeconfig file exists ensure permissions are set to 600 or more restrictive (Manual)

Test ID: 4.1.4 Status: WARN

4.1.4 If proxy kubeconfig file exists ensure ownership is set to root:root (Manual)

Test ID: 4.1.5 Status: PASS

4.1.5 Ensure that the --kubeconfig kubelet.conf file permissions are set to 600 or more restrictive (Automated)

Test ID: 4.1.6 Status: PASS

4.1.6 Ensure that the --kubeconfig kubelet.conf file ownership is set to root:root (Automated)

Test ID: 4.1.7 Status: WARN

4.1.7 Ensure that the certificate authorities file permissions are set to 600 or more restrictive (Manual)

Test ID: 4.1.8 Status: PASS

4.1.8 Ensure that the client certificate authorities file ownership is set to root:root (Manual)

Test ID: 4.1.9 Status: FAIL

4.1.9 If the kubelet config.yaml configuration file is being used validate permissions set to 600 or more restrictive (Automated)

Test ID: 4.1.10 Status: PASS

4.1.10 If the kubelet config.yaml configuration file is being used validate file ownership is set to root:root (Automated)

4.2 Kubelet

Test ID: 4.2.1 Status: PASS

4.2.1 Ensure that the --anonymous-auth argument is set to false (Automated)

Test ID: 4.2.2 Status: PASS

4.2.2 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Automated)

Test ID: 4.2.3 Status: PASS

4.2.3 Ensure that the --client-ca-file argument is set as appropriate (Automated)

Test ID: 4.2.4 Status: PASS

4.2.4 Verify that the --read-only-port argument is set to 0 (Manual)

Test ID: 4.2.5 Status: PASS

4.2.5 Ensure that the --streaming-connection-idle-timeout argument is not set to 0 (Manual)

Test ID: 4.2.6 Status: PASS

4.2.6 Ensure that the --make-iptables-util-chains argument is set to true (Automated)

Test ID: 4.2.7 Status: WARN

4.2.7 Ensure that the --hostname-override argument is not set (Manual)

Test ID: 4.2.8 Status: PASS

4.2.8 Ensure that the eventRecordQPS argument is set to a level which ensures appropriate event capture (Manual)

Test ID: 4.2.9 Status: WARN

4.2.9 Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate (Manual)

Test ID: 4.2.10 Status: PASS

4.2.10 Ensure that the --rotate-certificates argument is not set to false (Automated)

Test ID: 4.2.11 Status: PASS

4.2.11 Verify that the RotateKubeletServerCertificate argument is set to true (Manual)

Test ID: 4.2.12 Status: WARN

4.2.12 Ensure that the Kubelet only makes use of Strong Cryptographic Ciphers (Manual)

Test ID: 4.2.13 Status: WARN

4.2.13 Ensure that a limit is set on pod PIDs (Manual)

4.3 kube-proxy

Test ID: 4.3.1 Status: PASS

4.3.1 Ensure that the kube-proxy metrics service is bound to localhost (Automated)

== 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.d/10-kubeadm.conf

4.1.3 Run the below command (based on the file location on your system) on the each worker node.
For example,
chmod 600 /etc/kubernetes/proxy.conf

4.1.4 Run the below command (based on the file location on your system) on the each worker node.
For example, chown root:root /etc/kubernetes/proxy.conf

4.1.7 Run the following command to modify the file permissions of the
--client-ca-file chmod 600 <filename>

4.1.9 Run the following command (using the config file location identified in the Audit step)
chmod 600 /var/lib/kubelet/config.yaml

4.2.7 Edit the kubelet service file /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
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

4.2.9 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.d/10-kubeadm.conf on each worker node and
set the below parameters in KUBELET_CERTIFICATE_ARGS variable.
--tls-cert-file=<path/to/tls-certificate-file>
--tls-private-key-file=<path/to/tls-key-file>
Based on your system, restart the kubelet service. For example,
systemctl daemon-reload
systemctl restart kubelet.service

4.2.12 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_SHA256,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_GCM_SHA256
or to a subset of these values.
If using executable arguments, edit the kubelet service file
/etc/systemd/system/kubelet.service.d/10-kubeadm.conf 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_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_GCM_SHA256
Based on your system, restart the kubelet service. For example:
systemctl daemon-reload
systemctl restart kubelet.service

4.2.13 Decide on an appropriate level for this parameter and set it,
either via the --pod-max-pids command line parameter or the PodPidsLimit configuration file setting.

Summary node

Status	Count
PASS	15

FAIL	2
WARN	7
INFO	0

5 Kubernetes Policies

5.1 RBAC and Service Accounts

Test ID: 5.1.1 Status: FAIL

5.1.1 Ensure that the cluster-admin role is only used where required (Automated)

Test ID: 5.1.2 Status: FAIL

5.1.2 Minimize access to secrets (Automated)

Test ID: 5.1.3 Status: FAIL

5.1.3 Minimize wildcard use in Roles and ClusterRoles (Automated)

Test ID: 5.1.4 Status: FAIL

5.1.4 Minimize access to create pods (Automated)

Test ID: 5.1.5 Status: FAIL

5.1.5 Ensure that default service accounts are not actively used (Automated)

Test ID: 5.1.6 Status: FAIL

5.1.6 Ensure that Service Account Tokens are only mounted where necessary (Automated)

Test ID: 5.1.7 Status: WARN

5.1.7 Avoid use of system:masters group (Manual)

Test ID: 5.1.8 Status: WARN

5.1.8 Limit use of the Bind, Impersonate and Escalate permissions in the Kubernetes cluster (Manual)

Test ID: 5.1.9 Status: WARN

5.1.9 Minimize access to create persistent volumes (Manual)

Test ID: 5.1.10 Status: WARN

5.1.10 Minimize access to the proxy sub-resource of nodes (Manual)

Test ID: 5.1.11 Status: WARN

5.1.11 Minimize access to the approval sub-resource of certificatesigningrequests objects (Manual)

Test ID: 5.1.12 Status: WARN

5.1.12 Minimize access to webhook configuration objects (Manual)

Test ID: 5.1.13 Status: WARN

5.1.13 Minimize access to the service account token creation (Manual)

5.2 Pod Security Standards

Test ID: 5.2.1 Status: WARN

5.2.1 Ensure that the cluster has at least one active policy control mechanism in place (Manual)

Test ID: 5.2.2 Status: WARN

5.2.2 Minimize the admission of privileged containers (Manual)

Test ID: 5.2.3 Status: WARN

5.2.3 Minimize the admission of containers wishing to share the host process ID namespace (Manual)

Test ID: 5.2.4 Status: WARN

5.2.4 Minimize the admission of containers wishing to share the host IPC namespace (Manual)

Test ID: 5.2.5 Status: WARN

5.2.5 Minimize the admission of containers wishing to share the host network namespace (Manual)

Test ID: 5.2.6 Status: WARN

5.2.6 Minimize the admission of containers with allowPrivilegeEscalation (Manual)

Test ID: 5.2.7 Status: WARN

5.2.7 Minimize the admission of root containers (Manual)

Test ID: 5.2.8 Status: WARN

5.2.8 Minimize the admission of containers with the NET_RAW capability (Manual)

Test ID: 5.2.9 Status: WARN

5.2.9 Minimize the admission of containers with added capabilities (Manual)

Test ID: 5.2.10 Status: WARN

5.2.10 Minimize the admission of containers with capabilities assigned (Manual)

Test ID: 5.2.11 Status: WARN

5.2.11 Minimize the admission of Windows HostProcess containers (Manual)

Test ID: 5.2.12 Status: WARN

5.2.12 Minimize the admission of HostPath volumes (Manual)

Test ID: 5.2.13 Status: WARN

5.2.13 Minimize the admission of containers which use HostPorts (Manual)

5.3 Network Policies and CNI

Test ID: 5.3.1 Status: WARN

5.3.1 Ensure that the CNI in use supports NetworkPolicies (Manual)

Test ID: 5.3.2 Status: WARN

5.3.2 Ensure that all Namespaces have NetworkPolicies defined (Manual)

5.4 Secrets Management

Test ID: 5.4.1 Status: WARN

5.4.1 Prefer using Secrets as files over Secrets as environment variables (Manual)

Test ID: 5.4.2 Status: WARN

5.4.2 Consider external secret storage (Manual)

5.5 Extensible Admission Control

Test ID: 5.5.1 Status: WARN

5.5.1 Configure Image Provenance using ImagePolicyWebhook admission controller (Manual)

5.7 General Policies

Test ID: 5.7.1 Status: WARN

5.7.1 Create administrative boundaries between resources using namespaces (Manual)

Test ID: 5.7.2 Status: WARN

5.7.2 Ensure that the seccomp profile is set to docker/default in your Pod definitions (Manual)

Test ID: 5.7.3 Status: WARN

5.7.3 Apply SecurityContext to your Pods and Containers (Manual)

Test ID: 5.7.4 Status: 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]
Condition: is_compliant is false if rolename is not cluster-admin and rolebinding is cluster-admin.

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 roles and clusterroles with specific objects or actions.
Condition: role_is_compliant is false if ["*"] is found in rules.
Condition: clusterrole_is_compliant is false if ["*"] is found in rules.

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 ServiceAccounts and Pods which do not need to mount service account tokens to disable it, with `automountServiceAccountToken: false`.
If both the ServiceAccount and the Pod's .spec specify a value for automountServiceAccountToken, the Pod spec takes precedence.
Condition: Pod is_compliant to true when
- ServiceAccount is automountServiceAccountToken: false and Pod is automountServiceAccountToken: false or notset
- ServiceAccount is automountServiceAccountToken: true notset and Pod is automountServiceAccountToken: false

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.

5.1.9 Where possible, remove create access to PersistentVolume objects in the cluster.

5.1.10 Where possible, remove access to the proxy sub-resource of node objects.

5.1.11 Where possible, remove access to the approval sub-resource of certificatesigningrequests objects.

5.1.12 Where possible, remove access to the validatingwebhookconfigurations or mutatingwebhookconfigurations objects

5.1.13 Where possible, remove access to the token sub-resource of serviceaccount objects.

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.
Audit: the audit list all pods' containers to retrieve their .securityContext.privileged value.
Condition: is_compliant is false if container's `.securityContext.privileged` is set to `true`.
Default: by default, there are no restrictions on the creation 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.
Audit: the audit retrieves each Pod' spec.hostPID.

Condition: is_compliant is false if Pod's spec.hostPID is set to `true`.
Default: by default, there are no restrictions on the creation 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.
Audit: the audit retrieves each Pod' spec.IPC.
Condition: is_compliant is false if Pod's spec.hostIPC is set to `true`.
Default: by default, there are no restrictions on the creation 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.
Audit: the audit retrieves each Pod' spec.hostNetwork.
Condition: is_compliant is false if Pod's spec.hostNetwork is set to `true`.
Default: by default, there are no restrictions on the creation 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 `.securityContext.allowPrivilegeEscalation` set to `true`.
Audit: the audit retrieves each Pod's container(s) `.securityContext.allowPrivilegeEscalation`.
Condition: is_compliant is false if container's `.securityContext.allowPrivilegeEscalation` is set to `true`.
Default: If notset, privilege escalation is allowed (default to true). However if PSP/PSA is used with a `restricted` profile,
privilege escalation is explicitly disallowed unless configured otherwise.

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.
Audit: the audit retrieves each Pod's container(s) added capabilities.
Condition: is_compliant is false if added capabilities are added for a given container.
Default: Containers run with a default set of capabilities as assigned by the Container Runtime.

5.2.10 Review the use of capabilites 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.

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

Status	Count
PASS	0
FAIL	6
WARN	29
INFO	0