CIS v1.6 Benchmark - Self-Assessment Guide - Rancher v2.6

Contents

CIS v1.6 Kubernetes Benchmark - Rancher v2.6 with Kubernetes v1.18 to v1.21

Click here to download a PDF version of this document.

Overview

This document is a companion to the Rancher v2.6 security hardening guide. The hardening guide provides prescriptive guidance for hardening a production installation of Rancher, and this benchmark guide is meant to help you evaluate the level of security of the hardened cluster against each control in the benchmark.

This guide corresponds to specific versions of the hardening guide, Rancher, CIS Benchmark and Kubernetes:

Hardening Guide Version	Rancher Version	CIS Benchmark Version	Kubernetes Version
Hardening Guide CIS v1.6 Benchmark	Rancher v2.6.3	CIS v1.6	Kubernetes v1.18, v1.19, v1.20 and v1.21

Because Rancher and RKE install Kubernetes services as Docker containers, many of the control verification checks in the CIS Kubernetes Benchmark do not apply and will have a result of `Not Applicable`. This guide will walk through the various controls and provide updated example commands to audit compliance in Rancher created clusters.

This document is to be used by Rancher operators, security teams, auditors and decision makers.

For more detail about each audit, including rationales and remediations for failing tests, you can refer to the corresponding section of the CIS Kubernetes Benchmark v1.6. You can download the benchmark, after creating a free account, in Center for Internet Security (CIS).

Testing controls methodology

Rancher and RKE install Kubernetes services via Docker containers. Configuration is defined by arguments passed to the container at the time of initialization, not via configuration files.

Where control audits differ from the original CIS benchmark, the audit commands specific to Rancher are provided for testing. When performing the tests, you will need access to the Docker command line on the hosts of all three RKE roles. The commands also make use of the kubectl (with a valid configuration file) and jq tools, which are required in the testing and evaluation of test results.

NOTE: For the moment only `automated` tests (previously called `scored`) are covered in this guide.

Controls

1.1 Master Node Configuration Files

1.1.1 Ensure that the API server pod specification file permissions are set to 644 or more restrictive (Automated)

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for kube-apiserver. All configuration is passed in as arguments at container run time.

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

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for kube-apiserver. All configuration is passed in as arguments at container run time.

1.1.3 Ensure that the controller manager pod specification file permissions are set to 644 or more restrictive (Automated)

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for controller-manager. All configuration is passed in as arguments at container run time.

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

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for controller-manager. All configuration is passed in as arguments at container run time.

1.1.5 Ensure that the scheduler pod specification file permissions are set to 644 or more restrictive (Automated)

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for scheduler. All configuration is passed in as arguments at container run time.

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

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for scheduler. All configuration is passed in as arguments at container run time.

1.1.7 Ensure that the etcd pod specification file permissions are set to 644 or more restrictive (Automated)

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for etcd. All configuration is passed in as arguments at container run time.

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

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for etcd. All configuration is passed in as arguments at container run time.

1.1.9 Ensure that the Container Network Interface file permissions are set to 644 or more restrictive (Manual)

Result: warn

Remediation: Run the below command (based on the file location on your system) on the master node. For example, chmod 644

Audit:

```
stat -c permissions=%a <path/to/cni/files>
```

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

Result: warn

Remediation: Run the below command (based on the file location on your system) on the master node. For example, chown root:root

Audit:

```
stat -c %U:%G <path/to/cni/files>
```

1.1.13 Ensure that the admin.conf file permissions are set to 644 or more restrictive (Automated)

Result: Not Applicable

Remediation: Cluster provisioned by RKE does not store the Kubernetes default kubeconfig credentials file on the nodes.

1.1.14 Ensure that the admin.conf file ownership is set to root:root (Automated)

Result: Not Applicable

Remediation: Cluster provisioned by RKE does not store the Kubernetes default kubeconfig credentials file on the nodes.

1.1.15 Ensure that the scheduler.conf file permissions are set to 644 or more restrictive (Automated)

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for scheduler. All configuration is passed in as arguments at container run time.

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

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for scheduler. All configuration is passed in as arguments at container run time.

1.1.17 Ensure that the controller-manager.conf file permissions are set to 644 or more restrictive (Automated)

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for controller-manager. All configuration is passed in as arguments at container run time.

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

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn't require or maintain a configuration file for controller-manager. All configuration is passed in as arguments at container run time.

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

Result: pass

Remediation: Run the below command (based on the file location on your system) on the master node. For example, chown -R root:root /etc/kubernetes/pki/

Audit Script: `check_files_owner_in_dir.sh`

```
#!/usr/bin/env bash

# This script is used to ensure the owner is set to root:root
for
# the given directory and all the files in it
#
# inputs:
#   $1 = /full/path/to/directory
#
# outputs:
#   true/false

INPUT_DIR=$1

if [[ "${INPUT_DIR}" == "" ]]; then
    echo "false"
    exit
fi

if [[ $(stat -c %U:%G ${INPUT_DIR}) != "root:root" ]]; then
    echo "false"
    exit
fi
```

```
statInfoLines=$(stat -c "%n %U:%G" ${INPUT_DIR}/*)
while read -r statInfoLine; do
  f=$(echo ${statInfoLine} | cut -d' ' -f1)
  p=$(echo ${statInfoLine} | cut -d' ' -f2)

  if [[ $(basename "$f" .pem) == "kube-etcd-"* ]]; then
    if [[ "$p" != "root:root" && "$p" != "etcd:etcd" ]]; then
      echo "false"
      exit
    fi
  else
    if [[ "$p" != "root:root" ]]; then
      echo "false"
      exit
    fi
  fi
done <<< "${statInfoLines}"


echo "true"
exit
```

Audit Execution:

```
./check_files_owner_in_dir.sh /node/etc/kubernetes/ssl
```

Expected Result:

```
'true' is equal to 'true'
```

Returned Value:

```
true
```

1.1.20 Ensure that the Kubernetes PKI certificate file permissions are set to 644 or more restrictive (Automated)

Result: pass

Remediation: Run the below command (based on the file location on your system) on the master node. For example, chmod -R 644 /etc/kubernetes/pki/*.crt

Audit Script: `check_files_permissions.sh`

```
#!/usr/bin/env bash

# This script is used to ensure the file permissions are set
to 644 or
# more restrictive for all files in a given directory or a
wildcard
# selection of files
#
# inputs:
#   $1 = /full/path/to/directory or /path/to/fileswithpattern
#                                  ex: !(*key).pem
#
#   $2 (optional) = permission (ex: 600)
#
# outputs:
#   true/false

# Turn on "extended glob" for use of '!' in wildcard
shopt -s extglob

# Turn off history to avoid surprises when using '!'
set -H

USER_INPUT=$1

if [[ "${USER_INPUT}" == "" ]]; then
  echo "false"
  exit
fi



if [[ -d ${USER_INPUT} ]]; then
  PATTERN="${USER_INPUT}/*"
```

```
else
  PATTERN="${USER_INPUT}"
fi

PERMISSION=""
if [[ "$2" != "" ]]; then
  PERMISSION=$2
fi

FILES_PERMISSIONS=$(stat -c %n\ %a ${PATTERN})

while read -r fileInfo; do
  p=$(echo ${fileInfo} | cut -d' ' -f2)

  if [[ "${PERMISSION}" != "" ]]; then
    if [[ "$p" != "${PERMISSION}" ]]; then
      echo "false"
      exit
    fi
  else
    if [[ "$p" != "644" && "$p" != "640" && "$p" != "600" ]];
then
      echo "false"
      exit
    fi
  fi
done <<< "${FILES_PERMISSIONS}"


echo "true"
exit
```

Audit Execution:

```
./check_files_permissions.sh /node/etc/kubernetes/ssl/!(*key).
pem
```

Expected Result:

```
'true' is equal to 'true'
```

Returned Value:

```
true
```

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

Result: pass

Remediation: Run the below command (based on the file location on your system) on the master node. For example, chmod -R 600 /etc/kubernetes/ssl/*key.pem

Audit Script: `check_files_permissions.sh`

```
#!/usr/bin/env bash

# This script is used to ensure the file permissions are set
to 644 or
# more restrictive for all files in a given directory or a
wildcard
# selection of files
#
# inputs:
#   $1 = /full/path/to/directory or /path/to/fileswithpattern
#                                   ex: !(*key).pem
#
#   $2 (optional) = permission (ex: 600)
#
# outputs:
#   true/false

# Turn on "extended glob" for use of '!' in wildcard
shopt -s extglob

# Turn off history to avoid surprises when using '!'
set -H

USER_INPUT=$1
```

```
if [[ "${USER_INPUT}" == "" ]]; then
  echo "false"
  exit
fi


if [[ -d ${USER_INPUT} ]]; then
  PATTERN="${USER_INPUT}/*"
else
  PATTERN="${USER_INPUT}"
fi

PERMISSION=""
if [[ "$2" != "" ]]; then
  PERMISSION=$2
fi

FILES_PERMISSIONS=$(stat -c %n\ %a ${PATTERN})

while read -r fileInfo; do
  p=$(echo ${fileInfo} | cut -d' ' -f2)

  if [[ "${PERMISSION}" != "" ]]; then
    if [[ "$p" != "${PERMISSION}" ]]; then
      echo "false"
      exit
    fi
  else
    if [[ "$p" != "644" && "$p" != "640" && "$p" != "600" ]];
then
      echo "false"
      exit
    fi
  fi
done <<< "${FILES_PERMISSIONS}"
```

```
echo "true"
exit
```

Audit Execution:

```
./check_files_permissions.sh /node/etc/kubernetes/ssl/*key.pem
```

Expected Result:

```
'true' is equal to 'true'
```

Returned Value:

```
true
```

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

Result: pass

Remediation: On the etcd server node, get the etcd data directory, passed as an argument --data-dir, from the below command: ps -ef | grep etcd Run the below command (based on the etcd data directory found above). For example, chmod 700 /var/lib/etcd

Audit:

```
stat -c %a /node/var/lib/etcd
```

Expected Result:

```
'700' is equal to '700'
```

Returned Value:

```
700
```

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

Result: pass

Remediation: On the etcd server node, get the etcd data directory, passed as an argument --data-dir, from the below 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

A system service account is required for etcd data directory ownership. Refer to Rancher's hardening guide for more details on how to configure this ownership.

Audit:

```
stat -c %U:%G /node/var/lib/etcd
```

Expected Result:

```
'etcd:etcd' is present
```

Returned Value:

```
etcd:etcd
```

1.2 API Server

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

Result: pass

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

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'false' is equal to 'false'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
```

```
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.2 Ensure that the --basic-auth-file argument is not set (Automated)

Result: pass

Remediation: Follow the documentation and configure alternate mechanisms for authentication. Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and remove the --basic-auth-file= parameter.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--basic-auth-file' is not present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
```

```
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Follow the documentation and configure alternate mechanisms for authentication. Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and remove the --token-auth-file= parameter.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--token-auth-file' is not present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
```

```
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.4 Ensure that the --kubelet-https argument is set to true (Automated)

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and remove the --kubelet-https parameter.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--kubelet-https' is not present OR '--kubelet-https' is not
present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
```

```
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Follow the Kubernetes documentation and set up the TLS connection between the apiserver and kubelets. Then, edit API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the kubelet client certificate and key parameters as below. --kubelet-client-certificate= --kubelet-client-key=

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--kubelet-client-certificate' is present AND '--kubelet-
client-key' is present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
```

```
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: 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 master node and set the --kubelet-certificate-authority parameter to the path to the cert file for the certificate authority. --kubelet-certificate-authority=

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--kubelet-certificate-authority' is present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
```

```
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --authorization-mode parameter to values other than AlwaysAllow. One such example could be as below. --authorization-mode=RBAC

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'Node,RBAC' not have 'AlwaysAllow'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
```

```
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --authorization-mode parameter to a value that includes Node. --authorization-mode=Node,RBAC

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'Node,RBAC' has 'Node'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
```

```
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/ manifests/kube-apiserver.yaml on the master node and set the -- authorization-mode parameter to a value that includes RBAC, for example: --authorization-mode=Node,RBAC

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'Node,RBAC' has 'RBAC'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
```

```
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.10 Ensure that the admission control plugin EventRateLimit is set (Automated)

Result: pass

Remediation: 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=

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageC
lass,DefaultTolerationSeconds,MutatingAdmissionWebhook,Validat
ingAdmissionWebhook,ResourceQuota,NodeRestriction,Priority,Tai
ntNodesByCondition,PersistentVolumeClaimResize,PodSecurityPoli
cy,EventRateLimit' has 'EventRateLimit'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
```

```
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and either remove the --enable-admission-plugins parameter, or set it to a value that does not include AlwaysAdmit.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageC
lass,DefaultTolerationSeconds,MutatingAdmissionWebhook,Validat
```

```
ingAdmissionWebhook,ResourceQuota,NodeRestriction,Priority,Tai
ntNodesByCondition,PersistentVolumeClaimResize,PodSecurityPoli
cy,EventRateLimit' not have 'AlwaysAdmit' OR '--enable-
admission-plugins' is not present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
```

```
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: warn

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

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

1.2.13 Ensure that the admission control plugin SecurityContextDeny is set if PodSecurityPolicy is not used (Manual)

Result: warn

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --enable-admission-plugins parameter to include SecurityContextDeny, unless PodSecurityPolicy is already in place. --enable-admission-plugins=...,SecurityContextDeny,...

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

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

Result: pass

Remediation: Follow the documentation and create ServiceAccount objects as per your environment. Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and ensure that the --disable-admission-plugins parameter is set to a value that does not include ServiceAccount.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--disable-admission-plugins' is not present OR '--disable-
admission-plugins' is not present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
```

```
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --disable-admission-plugins parameter to ensure it does not include NamespaceLifecycle.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--disable-admission-plugins' is not present OR '--disable-
admission-plugins' is not present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
```

```
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.16 Ensure that the admission control plugin PodSecurityPolicy is set (Automated)

Result: pass

Remediation: Follow the documentation and create Pod Security Policy objects as per your environment. Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --enable-admission-plugins parameter to a value that includes PodSecurityPolicy: --enable-admission-plugins=...,PodSecurityPolicy,... Then restart the API Server.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageC
lass,DefaultTolerationSeconds,MutatingAdmissionWebhook,Validat
ingAdmissionWebhook,ResourceQuota,NodeRestriction,Priority,Tai
ntNodesByCondition,PersistentVolumeClaimResize,PodSecurityPoli
cy,EventRateLimit' has 'PodSecurityPolicy'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
```

```
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
```

```
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Follow the Kubernetes documentation and configure NodeRestriction plug-in on kubelets. Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --enable-admission-plugins parameter to a value that includes NodeRestriction. --enable-admission-plugins=...,NodeRestriction,...

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageC
lass,DefaultTolerationSeconds,MutatingAdmissionWebhook,Validat
ingAdmissionWebhook,ResourceQuota,NodeRestriction,Priority,Tai
ntNodesByCondition,PersistentVolumeClaimResize,PodSecurityPoli
cy,EventRateLimit' has 'NodeRestriction'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
```

```
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.18 Ensure that the --insecure-bind-address argument is not set (Automated)

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and remove the --insecure-bind-address parameter.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--insecure-bind-address' is not present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
```

```
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.19 Ensure that the --insecure-port argument is set to 0 (Automated)

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the below parameter. --insecure-port=0

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'0' is equal to '0'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
```

```
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
```

```
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.20 Ensure that the --secure-port argument is not set to 0 (Automated)

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and either remove the --secure-port parameter or set it to a different (non-zero) desired port.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
6443 is greater than 0 OR '--secure-port' is not present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
```

```
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

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

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'false' is equal to 'false'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
```

```
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master 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

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--audit-log-path' is present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
```

```
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --audit-log-maxage parameter to 30 or as an appropriate number of days: --audit-log-maxage=30

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
30 is greater or equal to 30
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
```

```
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --audit-log-maxbackup parameter to 10 or to an appropriate value. --audit-log-maxbackup=10

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
10 is greater or equal to 10
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
```

```
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master 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

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
100 is greater or equal to 100
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
```

```
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.26 Ensure that the --request-timeout argument is set as appropriate (Automated)

Result: pass

Remediation: 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

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--request-timeout' is not present OR '--request-timeout' is
not present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
```

```
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the below parameter. --service-account-lookup=true Alternatively, you can delete the --service-account-lookup parameter from this file so that the default takes effect.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--service-account-lookup' is not present OR 'true' is equal
to 'true'
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
```

```
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the --service-account-key-file parameter to the public key file for service accounts: --service-account-key-file=

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--service-account-key-file' is present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
```

```
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Follow the Kubernetes documentation and set up the TLS connection between the apiserver and etcd. Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the etcd certificate and key file parameters. --etcd-certfile= --etcd-keyfile=

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--etcd-certfile' is present AND '--etcd-keyfile' is present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
```

```
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Follow the Kubernetes documentation and set up the TLS connection on the apiserver. Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the TLS certificate and private key file parameters. --tls-cert-file= --tls-private-key-file=

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--tls-cert-file' is present AND '--tls-private-key-file' is
present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
```

```
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Follow the Kubernetes documentation and set up the TLS connection on the apiserver. Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the client certificate authority file. --client-ca-file=

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--client-ca-file' is present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
```

```
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: pass

Remediation: Follow the Kubernetes documentation and set up the TLS connection between the apiserver and etcd. Then, edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the etcd certificate authority file parameter. --etcd-cafile=

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--etcd-cafile' is present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
```

```
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.33 Ensure that the --encryption-provider-config argument is set as appropriate (Automated)

Result: pass

Remediation: 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 master node and set the --encryption-provider-config parameter to the path of that file: --encryption-provider-config=

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--encryption-provider-config' is present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:02 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
```

```
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
```

```
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

1.2.34 Ensure that encryption providers are appropriately configured (Automated)

Result: pass

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

Audit Script: `check_encryption_provider_config.sh`

```
#!/usr/bin/env bash

# This script is used to check the encrption provider config
is set to aesbc
#
# outputs:
#   true/false

# TODO: Figure out the file location from the kube-apiserver
commandline args
ENCRYPTION_CONFIG_FILE="/node/etc/kubernetes/ssl/
encryption.yaml"

if [[ ! -f "${ENCRYPTION_CONFIG_FILE}" ]]; then
  echo "false"
  exit
fi

for provider in "$@"
do
  if grep "$provider" "${ENCRYPTION_CONFIG_FILE}"; then
    echo "true"
    exit
    fi
done

echo "false"
exit
```

Audit Execution:

```
./check_encryption_provider_config.sh aescbc
```

Expected Result:

```
'true' is equal to 'true'
```

Returned Value:

```
- aescbc: true
```

1.2.35 Ensure that the API Server only makes use of Strong Cryptographic Ciphers (Automated)

Result: warn

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the master node and set the below parameter. --tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_SHA256,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305,TLS_ECDHE_RSA_WITH_AES
_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305,TLS_ECDHE_ECDSA_WITH_AES
_SHA384

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

1.3 Controller Manager

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

Result: pass

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

Audit:

```
/bin/ps -ef | grep kube-controller-manager | grep -v grep
```

Expected Result:

```
'--terminated-pod-gc-threshold' is present
```

Returned Value:

```
root 6684 6662 1 13:04 ? 00:00:12 kube-controller-manager --
profiling=false --cluster-cidr=10.42.0.0/16 --service-account-
private-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --service-cluster-ip-range=10.43.0.0/16 --
address=0.0.0.0 --leader-elect=true --node-monitor-grace-
period=40s --v=2 --allocate-node-cidrs=true --enable-hostpath-
provisioner=false --pod-eviction-timeout=5m0s --configure-
cloud-routes=false --feature-
gates=RotateKubeletServerCertificate=true --cloud-provider= --
kubeconfig=/etc/kubernetes/ssl/kubecfg-kube-controller-
manager.yaml --root-ca-file=/etc/kubernetes/ssl/kube-ca.pem --
terminated-pod-gc-threshold=1000 --allow-untagged-cloud=true
--use-service-account-credentials=true
```

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

Result: pass

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

Audit:

```
/bin/ps -ef | grep kube-controller-manager | grep -v grep
```

Expected Result:

```
'false' is equal to 'false'
```

Returned Value:

```
root 6684 6662 1 13:04 ? 00:00:12 kube-controller-manager --
profiling=false --cluster-cidr=10.42.0.0/16 --service-account-
private-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --service-cluster-ip-range=10.43.0.0/16 --
address=0.0.0.0 --leader-elect=true --node-monitor-grace-
period=40s --v=2 --allocate-node-cidrs=true --enable-hostpath-
provisioner=false --pod-eviction-timeout=5m0s --configure-
cloud-routes=false --feature-
gates=RotateKubeletServerCertificate=true --cloud-provider= --
kubeconfig=/etc/kubernetes/ssl/kubecfg-kube-controller-
manager.yaml --root-ca-file=/etc/kubernetes/ssl/kube-ca.pem --
terminated-pod-gc-threshold=1000 --allow-untagged-cloud=true
--use-service-account-credentials=true
```

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

Result: pass

Remediation: Edit the Controller Manager pod specification file /etc/kubernetes/manifests/kube-controller-manager.yaml on the master node to set the below parameter. --use-service-account-credentials=true

Audit:

```
/bin/ps -ef | grep kube-controller-manager | grep -v grep
```

Expected Result:

```
'true' is not equal to 'false'
```

Returned Value:

```
root 6684 6662 1 13:04 ? 00:00:12 kube-controller-manager --
profiling=false --cluster-cidr=10.42.0.0/16 --service-account-
private-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --service-cluster-ip-range=10.43.0.0/16 --
address=0.0.0.0 --leader-elect=true --node-monitor-grace-
period=40s --v=2 --allocate-node-cidrs=true --enable-hostpath-
provisioner=false --pod-eviction-timeout=5m0s --configure-
cloud-routes=false --feature-
gates=RotateKubeletServerCertificate=true --cloud-provider= --
kubeconfig=/etc/kubernetes/ssl/kubecfg-kube-controller-
manager.yaml --root-ca-file=/etc/kubernetes/ssl/kube-ca.pem --
terminated-pod-gc-threshold=1000 --allow-untagged-cloud=true
--use-service-account-credentials=true
```

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

Result: pass

Remediation: Edit the Controller Manager pod specification file /etc/kubernetes/manifests/kube-controller-manager.yaml on the master node and set the --service-account-private-key-file parameter to the private key file for service accounts. --service-account-private-key-file=

Audit:

```
/bin/ps -ef | grep kube-controller-manager | grep -v grep
```

Expected Result:

```
'--service-account-private-key-file' is present
```

Returned Value:

```
root 6684 6662 1 13:04 ? 00:00:12 kube-controller-manager --
profiling=false --cluster-cidr=10.42.0.0/16 --service-account-
private-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --service-cluster-ip-range=10.43.0.0/16 --
address=0.0.0.0 --leader-elect=true --node-monitor-grace-
period=40s --v=2 --allocate-node-cidrs=true --enable-hostpath-
provisioner=false --pod-eviction-timeout=5m0s --configure-
```

```
cloud-routes=false --feature-
gates=RotateKubeletServerCertificate=true --cloud-provider= --
kubeconfig=/etc/kubernetes/ssl/kubecfg-kube-controller-
manager.yaml --root-ca-file=/etc/kubernetes/ssl/kube-ca.pem --
terminated-pod-gc-threshold=1000 --allow-untagged-cloud=true
--use-service-account-credentials=true
```

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

Result: pass

Remediation: Edit the Controller Manager pod specification file /etc/kubernetes/manifests/kube-controller-manager.yaml on the master node and set the --root-ca-file parameter to the certificate bundle file`. --root-ca-file=

Audit:

```
/bin/ps -ef | grep kube-controller-manager | grep -v grep
```

Expected Result:

```
'--root-ca-file' is present
```

Returned Value:

```
root 6684 6662 1 13:04 ? 00:00:12 kube-controller-manager --
profiling=false --cluster-cidr=10.42.0.0/16 --service-account-
private-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --service-cluster-ip-range=10.43.0.0/16 --
address=0.0.0.0 --leader-elect=true --node-monitor-grace-
period=40s --v=2 --allocate-node-cidrs=true --enable-hostpath-
provisioner=false --pod-eviction-timeout=5m0s --configure-
cloud-routes=false --feature-
gates=RotateKubeletServerCertificate=true --cloud-provider= --
kubeconfig=/etc/kubernetes/ssl/kubecfg-kube-controller-
manager.yaml --root-ca-file=/etc/kubernetes/ssl/kube-ca.pem --
terminated-pod-gc-threshold=1000 --allow-untagged-cloud=true
--use-service-account-credentials=true
```

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

Result: Not Applicable

Remediation: Edit the Controller Manager pod specification file /etc/kubernetes/manifests/kube-controller-manager.yaml on the master node and set the --feature-gates parameter to include RotateKubeletServerCertificate=true. --feature-gates=RotateKubeletServerCertificate=true

Cluster provisioned by RKE handles certificate rotation directly through RKE.

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

Result: pass

Remediation: Edit the Controller Manager pod specification file /etc/kubernetes/manifests/kube-controller-manager.yaml on the master node and ensure the correct value for the --bind-address parameter

Audit:

```
/bin/ps -ef | grep kube-controller-manager | grep -v grep
```

Expected Result:

```
'--bind-address' is not present OR '--bind-address' is not
present
```

Returned Value:

```
root 6684 6662 1 13:04 ? 00:00:12 kube-controller-manager --
profiling=false --cluster-cidr=10.42.0.0/16 --service-account-
private-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --service-cluster-ip-range=10.43.0.0/16 --
address=0.0.0.0 --leader-elect=true --node-monitor-grace-
period=40s --v=2 --allocate-node-cidrs=true --enable-hostpath-
provisioner=false --pod-eviction-timeout=5m0s --configure-
cloud-routes=false --feature-
gates=RotateKubeletServerCertificate=true --cloud-provider= --
kubeconfig=/etc/kubernetes/ssl/kubecfg-kube-controller-
manager.yaml --root-ca-file=/etc/kubernetes/ssl/kube-ca.pem --
```

```
terminated-pod-gc-threshold=1000 --allow-untagged-cloud=true
--use-service-account-credentials=true
```

1.4 Scheduler

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

Result: pass

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

Audit:

```
/bin/ps -ef | grep kube-scheduler | grep -v grep
```

Expected Result:

```
'false' is equal to 'false'
```

Returned Value:

```
root 6889 6870 0 13:04 ? 00:00:02 kube-scheduler --leader-
elect=true --profiling=false --v=2 --kubeconfig=/etc/
kubernetes/ssl/kubecfg-kube-scheduler.yaml --address=0.0.0.0
```

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

Result: pass

Remediation: Edit the Scheduler pod specification file /etc/kubernetes/manifests/kube-scheduler.yaml on the master node and ensure the correct value for the --bind-address parameter

Audit:

```
/bin/ps -ef | grep kube-scheduler | grep -v grep
```

Expected Result:

```
'--bind-address' is not present OR '--bind-address' is not
present
```

Returned Value:

```
root 6889 6870 0 13:04 ? 00:00:02 kube-scheduler --leader-
elect=true --profiling=false --v=2 --kubeconfig=/etc/
kubernetes/ssl/kubecfg-kube-scheduler.yaml --address=0.0.0.0
```

2 Etcd Node Configuration Files

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

Result: pass

Remediation: Follow the etcd service documentation and configure TLS encryption. Then, edit the etcd pod specification file /etc/kubernetes/manifests/etcd.yaml on the master node and set the below parameters. --cert-file= --key-file=

Audit:

```
/bin/ps -ef | /bin/grep etcd | /bin/grep -v grep
```

Expected Result:

```
'--cert-file' is present AND '--key-file' is present
```

Returned Value:

```
etcd 6259 6237 1 13:03 ? 00:00:13 /usr/local/bin/etcd --
listen-peer-urls=https://<node_ip>:2380 --peer-trusted-ca-
file=/etc/kubernetes/ssl/kube-ca.pem --peer-cert-file=/etc/
kubernetes/ssl/kube-etcd-<node_ip>.pem --peer-client-cert-
auth=true --heartbeat-interval=500 --name=etcd-<external_ip>
--initial-cluster=etcd-<external_ip>=https://<node_ip>:2380 --
trusted-ca-file=/etc/kubernetes/ssl/kube-ca.pem --peer-key-
file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --cipher-
suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WIT
H_AES_256_GCM_SHA384 --listen-client-urls=https://<node_ip>:
2379 --initial-advertise-peer-urls=https://<node_ip>:2380 --
key-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --
client-cert-auth=true --enable-v2=true --election-
timeout=5000 --data-dir=/var/lib/rancher/etcd/ --initial-
cluster-token=etcd-cluster-1 --initial-cluster-state=new --
cert-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>.pem --
advertise-client-urls=https://<node_ip>:2379 root 6465 6444 8
```

```
13:04 ? 00:01:02 kube-apiserver --requestheader-allowed-
names=kube-apiserver-proxy-client --etcd-cafile=/etc/
kubernetes/ssl/kube-ca.pem --service-node-port-
range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
```

```
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10 root 24183 24165 1
13:16 ? 00:00:00 kube-bench run --targets etcd --scored --
nosummary --noremediations --v=5 --config-dir=/etc/kube-bench/
cfg --benchmark rke-cis-1.6-hardened --json --log_dir /tmp/
results/logs --outputfile /tmp/results/etcd.json
```

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

Result: pass

Remediation: Edit the etcd pod specification file /etc/kubernetes/manifests/etcd.yaml on the master node and set the below parameter. --client-cert-auth="true"

Audit:

```
/bin/ps -ef | /bin/grep etcd | /bin/grep -v grep
```

Expected Result:

```
'--client-cert-auth' is present OR 'true' is equal to 'true'
```

Returned Value:

```
etcd 6259 6237 1 13:03 ? 00:00:13 /usr/local/bin/etcd --
listen-peer-urls=https://<node_ip>:2380 --peer-trusted-ca-
file=/etc/kubernetes/ssl/kube-ca.pem --peer-cert-file=/etc/
kubernetes/ssl/kube-etcd-<node_ip>.pem --peer-client-cert-
auth=true --heartbeat-interval=500 --name=etcd-<external_ip>
--initial-cluster=etcd-<external_ip>=https://<node_ip>:2380 --
trusted-ca-file=/etc/kubernetes/ssl/kube-ca.pem --peer-key-
file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --cipher-
suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WIT
H_AES_256_GCM_SHA384 --listen-client-urls=https://<node_ip>:
2379 --initial-advertise-peer-urls=https://<node_ip>:2380 --
```

```
key-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --
client-cert-auth=true --enable-v2=true --election-
timeout=5000 --data-dir=/var/lib/rancher/etcd/ --initial-
cluster-token=etcd-cluster-1 --initial-cluster-state=new --
cert-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>.pem --
advertise-client-urls=https://<node_ip>:2379 root 6465 6444 8
13:04 ? 00:01:02 kube-apiserver --requestheader-allowed-
names=kube-apiserver-proxy-client --etcd-cafile=/etc/
kubernetes/ssl/kube-ca.pem --service-node-port-
range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
```

```
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10 root 24183 24165 1
13:16 ? 00:00:00 kube-bench run --targets etcd --scored --
nosummary --noremediations --v=5 --config-dir=/etc/kube-bench/
cfg --benchmark rke-cis-1.6-hardened --json --log_dir /tmp/
results/logs --outputfile /tmp/results/etcd.json
```

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

Result: pass

Remediation: Edit the etcd pod specification file /etc/kubernetes/manifests/etcd.yaml on the master node and either remove the --auto-tls parameter or set it to false. --auto-tls=false

Audit:

```
/bin/ps -ef | /bin/grep etcd | /bin/grep -v grep
```

Expected Result:

```
'--auto-tls' is not present OR '--auto-tls' is not present
```

Returned Value:

```
etcd 6259 6237 1 13:03 ? 00:00:13 /usr/local/bin/etcd --
listen-peer-urls=https://<node_ip>:2380 --peer-trusted-ca-
file=/etc/kubernetes/ssl/kube-ca.pem --peer-cert-file=/etc/
kubernetes/ssl/kube-etcd-<node_ip>.pem --peer-client-cert-
auth=true --heartbeat-interval=500 --name=etcd-<external_ip>
```

```
--initial-cluster=etcd-<external_ip>=https://<node_ip>:2380 --
trusted-ca-file=/etc/kubernetes/ssl/kube-ca.pem --peer-key-
file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --cipher-
suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WIT
H_AES_256_GCM_SHA384 --listen-client-urls=https://<node_ip>:
2379 --initial-advertise-peer-urls=https://<node_ip>:2380 --
key-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --
client-cert-auth=true --enable-v2=true --election-
timeout=5000 --data-dir=/var/lib/rancher/etcd/ --initial-
cluster-token=etcd-cluster-1 --initial-cluster-state=new --
cert-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>.pem --
advertise-client-urls=https://<node_ip>:2379 root 6465 6444 8
13:04 ? 00:01:02 kube-apiserver --requestheader-allowed-
names=kube-apiserver-proxy-client --etcd-cafile=/etc/
kubernetes/ssl/kube-ca.pem --service-node-port-
range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
```

```
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10 root 24183 24165 1
13:16 ? 00:00:00 kube-bench run --targets etcd --scored --
nosummary --noremediations --v=5 --config-dir=/etc/kube-bench/
cfg --benchmark rke-cis-1.6-hardened --json --log_dir /tmp/
results/logs --outputfile /tmp/results/etcd.json
```

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

Result: pass

Remediation: Follow the etcd service documentation and configure peer TLS encryption as appropriate for your etcd cluster. Then, edit the etcd pod specification file /etc/kubernetes/manifests/etcd.yaml on the master node and set the below parameters. --peer-client-file= --peer-key-file=

Audit:

```
/bin/ps -ef | /bin/grep etcd | /bin/grep -v grep
```

Expected Result:

```
'--peer-cert-file' is present AND '--peer-key-file' is present
```

Returned Value:

```
etcd 6259 6237 1 13:03 ? 00:00:13 /usr/local/bin/etcd --
listen-peer-urls=https://<node_ip>:2380 --peer-trusted-ca-
file=/etc/kubernetes/ssl/kube-ca.pem --peer-cert-file=/etc/
kubernetes/ssl/kube-etcd-<node_ip>.pem --peer-client-cert-
auth=true --heartbeat-interval=500 --name=etcd-<external_ip>
--initial-cluster=etcd-<external_ip>=https://<node_ip>:2380 --
trusted-ca-file=/etc/kubernetes/ssl/kube-ca.pem --peer-key-
file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --cipher-
suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WIT
H_AES_256_GCM_SHA384 --listen-client-urls=https://<node_ip>:
2379 --initial-advertise-peer-urls=https://<node_ip>:2380 --
key-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --
client-cert-auth=true --enable-v2=true --election-
timeout=5000 --data-dir=/var/lib/rancher/etcd/ --initial-
cluster-token=etcd-cluster-1 --initial-cluster-state=new --
cert-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>.pem --
advertise-client-urls=https://<node_ip>:2379 root 6465 6444 8
13:04 ? 00:01:02 kube-apiserver --requestheader-allowed-
names=kube-apiserver-proxy-client --etcd-cafile=/etc/
kubernetes/ssl/kube-ca.pem --service-node-port-
range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
```

```
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10 root 24183 24165 1
13:16 ? 00:00:00 kube-bench run --targets etcd --scored --
nosummary --noremediations --v=5 --config-dir=/etc/kube-bench/
cfg --benchmark rke-cis-1.6-hardened --json --log_dir /tmp/
results/logs --outputfile /tmp/results/etcd.json
```

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

Result: pass

Remediation: Edit the etcd pod specification file /etc/kubernetes/manifests/etcd.yaml on the master node and set the below parameter.
--peer-client-cert-auth=true

Audit:

```
/bin/ps -ef | /bin/grep etcd | /bin/grep -v grep
```

Expected Result:

```
'--peer-client-cert-auth' is present OR 'true' is equal to
'true'
```

Returned Value:

```
etcd 6259 6237 1 13:03 ? 00:00:13 /usr/local/bin/etcd --
listen-peer-urls=https://<node_ip>:2380 --peer-trusted-ca-
file=/etc/kubernetes/ssl/kube-ca.pem --peer-cert-file=/etc/
kubernetes/ssl/kube-etcd-<node_ip>.pem --peer-client-cert-
auth=true --heartbeat-interval=500 --name=etcd-<external_ip>
--initial-cluster=etcd-<external_ip>=https://<node_ip>:2380 --
trusted-ca-file=/etc/kubernetes/ssl/kube-ca.pem --peer-key-
file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --cipher-
suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WIT
H_AES_256_GCM_SHA384 --listen-client-urls=https://<node_ip>:
2379 --initial-advertise-peer-urls=https://<node_ip>:2380 --
key-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --
client-cert-auth=true --enable-v2=true --election-
timeout=5000 --data-dir=/var/lib/rancher/etcd/ --initial-
cluster-token=etcd-cluster-1 --initial-cluster-state=new --
cert-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>.pem --
advertise-client-urls=https://<node_ip>:2379 root 6465 6444 8
13:04 ? 00:01:02 kube-apiserver --requestheader-allowed-
names=kube-apiserver-proxy-client --etcd-cafile=/etc/
kubernetes/ssl/kube-ca.pem --service-node-port-
range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
```

```
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10 root 24183 24165 1
13:16 ? 00:00:00 kube-bench run --targets etcd --scored --
```

```
nosummary --noremediations --v=5 --config-dir=/etc/kube-bench/
cfg --benchmark rke-cis-1.6-hardened --json --log_dir /tmp/
results/logs --outputfile /tmp/results/etcd.json
```

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

Result: pass

Remediation: Edit the etcd pod specification file /etc/kubernetes/manifests/etcd.yaml on the master node and either remove the --peer-auto-tls parameter or set it to false. --peer-auto-tls=false

Audit:

```
/bin/ps -ef | /bin/grep etcd | /bin/grep -v grep
```

Expected Result:

```
'--peer-auto-tls' is not present OR '--peer-auto-tls' is
present
```

Returned Value:

```
etcd 6259 6237 1 13:03 ? 00:00:13 /usr/local/bin/etcd --
listen-peer-urls=https://<node_ip>:2380 --peer-trusted-ca-
file=/etc/kubernetes/ssl/kube-ca.pem --peer-cert-file=/etc/
kubernetes/ssl/kube-etcd-<node_ip>.pem --peer-client-cert-
auth=true --heartbeat-interval=500 --name=etcd-<external_ip>
--initial-cluster=etcd-<external_ip>=https://<node_ip>:2380 --
trusted-ca-file=/etc/kubernetes/ssl/kube-ca.pem --peer-key-
file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --cipher-
suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WIT
H_AES_256_GCM_SHA384 --listen-client-urls=https://<node_ip>:
2379 --initial-advertise-peer-urls=https://<node_ip>:2380 --
key-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --
client-cert-auth=true --enable-v2=true --election-
timeout=5000 --data-dir=/var/lib/rancher/etcd/ --initial-
cluster-token=etcd-cluster-1 --initial-cluster-state=new --
cert-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>.pem --
advertise-client-urls=https://<node_ip>:2379 root 6465 6444 8
13:04 ? 00:01:02 kube-apiserver --requestheader-allowed-
names=kube-apiserver-proxy-client --etcd-cafile=/etc/
```

```
kubernetes/ssl/kube-ca.pem --service-node-port-
range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
```

```
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10 root 24183 24165 2
13:16 ? 00:00:00 kube-bench run --targets etcd --scored --
nosummary --noremediations --v=5 --config-dir=/etc/kube-bench/
cfg --benchmark rke-cis-1.6-hardened --json --log_dir /tmp/
results/logs --outputfile /tmp/results/etcd.json
```

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

Result: pass

Remediation: [Manual test] Follow the etcd documentation and create a dedicated certificate authority setup for the etcd service. Then, edit the etcd pod specification file /etc/kubernetes/manifests/etcd.yaml on the master node and set the below parameter. --trusted-ca-file=

Audit:

```
/bin/ps -ef | /bin/grep etcd | /bin/grep -v grep
```

Expected Result:

```
'--trusted-ca-file' is present
```

Returned Value:

```
etcd 6259 6237 1 13:03 ? 00:00:13 /usr/local/bin/etcd --
listen-peer-urls=https://<node_ip>:2380 --peer-trusted-ca-
file=/etc/kubernetes/ssl/kube-ca.pem --peer-cert-file=/etc/
kubernetes/ssl/kube-etcd-<node_ip>.pem --peer-client-cert-
auth=true --heartbeat-interval=500 --name=etcd-<external_ip>
--initial-cluster=etcd-<external_ip>=https://<node_ip>:2380 --
trusted-ca-file=/etc/kubernetes/ssl/kube-ca.pem --peer-key-
file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --cipher-
suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WIT
H_AES_256_GCM_SHA384 --listen-client-urls=https://<node_ip>:
2379 --initial-advertise-peer-urls=https://<node_ip>:2380 --
key-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>-key.pem --
```

```
client-cert-auth=true --enable-v2=true --election-
timeout=5000 --data-dir=/var/lib/rancher/etcd/ --initial-
cluster-token=etcd-cluster-1 --initial-cluster-state=new --
cert-file=/etc/kubernetes/ssl/kube-etcd-<node_ip>.pem --
advertise-client-urls=https://<node_ip>:2379 root 6465 6444 8
13:04 ? 00:01:02 kube-apiserver --requestheader-allowed-
names=kube-apiserver-proxy-client --etcd-cafile=/etc/
kubernetes/ssl/kube-ca.pem --service-node-port-
range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
```

```
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10 root 24183 24165 2
13:16 ? 00:00:00 kube-bench run --targets etcd --scored --
nosummary --noremediations --v=5 --config-dir=/etc/kube-bench/
cfg --benchmark rke-cis-1.6-hardened --json --log_dir /tmp/
results/logs --outputfile /tmp/results/etcd.json
```

3.1 Authentication and Authorization

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

Result: warn

Remediation: Alternative mechanisms provided by Kubernetes such as the use of OIDC should be implemented in place of client certificates.

3.2 Logging

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

Result: pass

Remediation: Create an audit policy file for your cluster.

Audit:

```
/bin/ps -ef | grep kube-apiserver | grep -v grep
```

Expected Result:

```
'--audit-policy-file' is present
```

Returned Value:

```
root 6465 6444 8 13:04 ? 00:01:03 kube-apiserver --
requestheader-allowed-names=kube-apiserver-proxy-client --
etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem --service-node-
port-range=30000-32767 --encryption-provider-config=/etc/
kubernetes/ssl/encryption.yaml --secure-port=6443 --proxy-
client-cert-file=/etc/kubernetes/ssl/kube-apiserver-proxy-
client.pem --tls-cert-file=/etc/kubernetes/ssl/kube-
apiserver.pem --api-audiences=unknown --storage-backend=etcd3
--tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA
_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY130
5,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES
_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305 --audit-
policy-file=/etc/kubernetes/audit-policy.yaml --etcd-
servers=https://<node_ip>:2379 --kubelet-certificate-
authority=/etc/kubernetes/ssl/kube-ca.pem --service-account-
signing-key-file=/etc/kubernetes/ssl/kube-service-account-
token-key.pem --admission-control-config-file=/etc/kubernetes/
admission.yaml --etcd-prefix=/registry --service-account-key-
file=/etc/kubernetes/ssl/kube-service-account-token-key.pem --
tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-
```

```
key.pem --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --advertise-
address=<node_ip> --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --requestheader-client-ca-file=/etc/kubernetes/ssl/
kube-apiserver-requestheader-ca.pem --requestheader-extra-
headers-prefix=X-Remote-Extra- --enable-admission-
plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultS
torageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,
ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Prior
ity,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecur
ityPolicy,EventRateLimit --authorization-mode=Node,RBAC --
cloud-provider= --etcd-certfile=/etc/kubernetes/ssl/kube-
node.pem --etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem
--service-cluster-ip-range=10.43.0.0/16 --profiling=false --
service-account-issuer=rke --allow-privileged=true --insecure-
port=0 --anonymous-auth=false --audit-log-path=/var/log/kube-
audit/audit-log.json --kubelet-client-certificate=/etc/
kubernetes/ssl/kube-apiserver.pem --kubelet-client-key=/etc/
kubernetes/ssl/kube-apiserver-key.pem --proxy-client-key-
file=/etc/kubernetes/ssl/kube-apiserver-proxy-client-key.pem
--bind-address=0.0.0.0 --kubelet-preferred-address-
types=InternalIP,ExternalIP,Hostname --audit-log-maxsize=100
--audit-log-format=json --service-account-lookup=true --
runtime-config=policy/v1beta1/podsecuritypolicy=true --audit-
log-maxage=30 --audit-log-maxbackup=10
```

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

Result: warn

Remediation: Consider modification of the audit policy in use on the cluster to include these items, at a minimum.

4.1 Worker Node Configuration Files

4.1.1 Ensure that the kubelet service file permissions are set to 644 or more restrictive (Automated)

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn’t require or maintain a configuration file for the kubelet service. All configuration is passed in as arguments at container run time.

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

Result: Not Applicable

Remediation: Cluster provisioned by RKE doesn’t require or maintain a configuration file for the kubelet service. All configuration is passed in as arguments at container run time.

4.1.3 If proxy kubeconfig file exists ensure permissions are set to 644 or more restrictive (Automated)

Result: pass

Remediation: Run the below command (based on the file location on your system) on the each worker node. For example, chmod 644 $proykubeconfig

Audit:

```
/bin/sh -c 'if test -e /node/etc/kubernetes/ssl/kubecfg-kube-proxy.yaml; then stat -c %a /node/etc/kubernetes/ssl/kubecfg-kube-proxy.yaml; fi'
```

Expected Result:

```
'644' is present OR '640' is present OR '600' is equal to '600' OR '444' is present OR '440' is present OR '400' is present OR '000' is present
```

Returned Value:

```
600
```

4.1.4 Ensure that the proxy kubeconfig file ownership is set to root:root (Automated)

Result: pass

Remediation: Run the below command (based on the file location on your system) on the each worker node. For example, chown root:root /etc/kubernetes/ssl/kubecfg-kube-proxy.yaml

Audit:

```
/bin/sh -c 'if test -e /etc/kubernetes/ssl/kubecfg-kube-proxy.yaml; then stat -c %U:%G /etc/kubernetes/ssl/kubecfg-kube-proxy.yaml; fi'
```

Expected Result:

```
'root:root' is not present OR '/etc/kubernetes/ssl/kubecfg-kube-proxy.yaml' is not present
```

4.1.5 Ensure that the --kubeconfig kubelet.conf file permissions are set to 644 or more restrictive (Automated)

Result: pass

Remediation: Run the below command (based on the file location on your system) on the each worker node. For example, chmod 644 /etc/kubernetes/ssl/kubecfg-kube-node.yaml

Audit:

```
/bin/sh -c 'if test -e /etc/kubernetes/ssl/kubecfg-kube-node.yaml; then stat -c permissions=%a /etc/kubernetes/ssl/kubecfg-kube-node.yaml; fi'
```

Expected Result:

```
'permissions' is not present
```

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

Result: pass

Remediation: Run the below command (based on the file location on your system) on the each worker node. For example, chown root:root /etc/kubernetes/ssl/kubecfg-kube-node.yaml

Audit:

```
/bin/sh -c 'if test -e /node/etc/kubernetes/ssl/kubecfg-kube-
node.yaml; then stat -c %U:%G /node/etc/kubernetes/ssl/
kubecfg-kube-node.yaml; fi'
```

Expected Result:

```
'root:root' is equal to 'root:root'
```

Returned Value:

```
root:root
```

4.1.7 Ensure that the certificate authorities file permissions are set to 644 or more restrictive (Automated)

Result: pass

Remediation: Run the following command to modify the file permissions of the --client-ca-file chmod 644

Audit Script: `check_cafile_permissions.sh`

```
#!/usr/bin/env bash

CAFILE=$(ps -ef | grep kubelet | grep -v apiserver | grep --
--client-ca-file= | awk -F '--client-ca-file=' '{print $2}' |
awk '{print $1}')
if test -z $CAFILE; then CAFILE=$kubeletcafile; fi
if test -e $CAFILE; then stat -c permissions=%a $CAFILE; fi
```

Audit Execution:

```
./check_cafile_permissions.sh
```

Expected Result:

```
'permissions' is not present
```

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

Result: pass

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

Audit Script: `check_cafile_ownership.sh`

```
#!/usr/bin/env bash

CAFILE=$(ps -ef | grep kubelet | grep -v apiserver | grep --
--client-ca-file= | awk -F '--client-ca-file=' '{print $2}' |
awk '{print $1}')
if test -z $CAFILE; then CAFILE=$kubeletcafile; fi
if test -e $CAFILE; then stat -c %U:%G $CAFILE; fi
```

Audit Execution:

```
./check_cafile_ownership.sh
```

Expected Result:

```
'root:root' is not present
```

4.1.9 Ensure that the kubelet --config configuration file has permissions set to 644 or more restrictive (Automated)

Result: Not Applicable

Remediation: Run the following command (using the config file location identified in the Audit step) chmod 644 /var/lib/kubelet/config.yaml

Clusters provisioned by RKE doesn't require or maintain a configuration file for the kubelet. All configuration is passed in as arguments at container run time.

4.1.10 Ensure that the kubelet --config configuration file ownership is set to root:root (Automated)

Result: Not Applicable

Remediation: Run the following command (using the config file location identified in the Audit step) chown root:root /var/lib/kubelet/config.yaml

Clusters provisioned by RKE doesn’t require or maintain a configuration file for the kubelet. All configuration is passed in as arguments at container run time.

4.2 Kubelet

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

Result: pass

Remediation: 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.d/10-kubeadm.conf 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

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'' is not present
```

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

Result: pass

Remediation: 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.d/10-kubeadm.conf 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

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'' is not present
```

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

Result: pass

Remediation: 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.d/10-kubeadm.conf 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

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'' is not present
```

4.2.4 Ensure that the --read-only-port argument is set to 0 (Automated)

Result: pass

Remediation: If using a Kubelet config file, edit the file to set readOnlyPort to 0. 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 parameter in KUBELET_SYSTEM_PODS_ARGS variable. --read-only-port=0 Based on your system, restart the kubelet service. For example: systemctl daemon-reload systemctl restart kubelet.service

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'' is not present OR '' is not present
```

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

Result: pass

Remediation: If using a Kubelet config file, edit the file to set streamingConnectionIdleTimeout to a value other than 0. 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 parameter in KUBELET_SYSTEM_PODS_ARGS variable. --streaming-connection-idle-timeout=5m Based on your system, restart the kubelet service. For example: systemctl daemon-reload systemctl restart kubelet.service

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'30m' is not equal to '0' OR '--streaming-connection-idle-
timeout' is not present
```

Returned Value:

```
UID PID PPID C STIME TTY TIME CMD root 7101 7078 3 13:04 ?
00:00:23 kubelet --streaming-connection-idle-timeout=30m --
cluster-dns=10.43.0.10 --pod-infra-container-image=rancher/
mirrored-pause:3.4.1 --node-ip=<node_ip> --network-plugin=cni
--event-qps=0 --kubeconfig=/etc/kubernetes/ssl/kubecfg-kube-
node.yaml --root-dir=/var/lib/kubelet --cni-bin-dir=/opt/cni/
bin --tls-cipher-
```

```
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_W
ITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACH
A20_POLY1305,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_RSA_W
ITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_GCM_SHA256 --v=2
--feature-gates=RotateKubeletServerCertificate=true --cloud-
provider= --hostname-override=<external_ip> --tls-cert-file=/
etc/kubernetes/ssl/kube-kubelet-<node_ip>.pem --authorization-
mode=Webhook --resolv-conf=/etc/resolv.conf --volume-plugin-
dir=/var/lib/kubelet/volumeplugins --cluster-
domain=cluster.local --tls-private-key-file=/etc/kubernetes/
ssl/kube-kubelet-<node_ip>-key.pem --authentication-token-
webhook=true --cni-conf-dir=/etc/cni/net.d --cgroups-per-
qos=True --make-iptables-util-chains=true --read-only-port=0
--fail-swap-on=false --anonymous-auth=false --protect-kernel-
defaults=true --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --address=0.0.0.0 --cgroup-driver=cgroupfs --resolv-
conf=/run/systemd/resolve/resolv.conf
```

4.2.6 Ensure that the --protect-kernel-defaults argument is set to true (Automated)

Result: pass

Remediation: If using a Kubelet config file, edit the file to set protectKernelDefaults: true. 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 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

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'' is not present
```

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

Result: pass

Remediation: If using a Kubelet config file, edit the file to set makeIPTablesUtilChains: true. If using command line arguments, edit the kubelet service file /etc/systemd/system/kubelet.service.d/10-kubeadm.conf on each worker node and remove the --make-iptables-util-chains 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

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'' is not present OR '' is not present
```

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

Result: Not Applicable

Remediation: 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

Clusters provisioned by RKE set the --hostname-override to avoid any hostname configuration errors

4.2.9 Ensure that the --event-qps argument is set to 0 or a level which ensures appropriate event capture (Automated)

Result: pass

Remediation: 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.d/10-kubeadm.conf 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

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'' is not present
```

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

Result: pass

Remediation: 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= --tls-private-key-file= Based on your system, restart the kubelet service. For example: systemctl daemon-reload systemctl restart kubelet.service

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'' is not present AND '' is not present
```

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

Result: pass

Remediation: If using a Kubelet config file, edit the file to add the line rotateCertificates: true or remove it altogether to use the default value. If using command line arguments, edit the kubelet service file /etc/systemd/system/kubelet.service.d/10-kubeadm.conf on each worker node and remove --rotate-certificates=false argument from the KUBELET_CERTIFICATE_ARGS variable. Based on your system, restart the kubelet service. For example: systemctl daemon-reload systemctl restart kubelet.service

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'--rotate-certificates' is not present OR '--rotate-
certificates' is not present
```

Returned Value:

```
UID PID PPID C STIME TTY TIME CMD root 7101 7078 3 13:04 ?
00:00:23 kubelet --streaming-connection-idle-timeout=30m --
cluster-dns=10.43.0.10 --pod-infra-container-image=rancher/
mirrored-pause:3.4.1 --node-ip=<node_ip> --network-plugin=cni
--event-qps=0 --kubeconfig=/etc/kubernetes/ssl/kubecfg-kube-
node.yaml --root-dir=/var/lib/kubelet --cni-bin-dir=/opt/cni/
bin --tls-cipher-
suites=TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_W
ITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACH
A20_POLY1305,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_RSA_W
ITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_GCM_SHA256 --v=2
--feature-gates=RotateKubeletServerCertificate=true --cloud-
provider= --hostname-override=<external_ip> --tls-cert-file=/
etc/kubernetes/ssl/kube-kubelet-<node_ip>.pem --authorization-
mode=Webhook --resolv-conf=/etc/resolv.conf --volume-plugin-
dir=/var/lib/kubelet/volumeplugins --cluster-
domain=cluster.local --tls-private-key-file=/etc/kubernetes/
ssl/kube-kubelet-<node_ip>-key.pem --authentication-token-
```

```
webhook=true --cni-conf-dir=/etc/cni/net.d --cgroups-per-
qos=True --make-iptables-util-chains=true --read-only-port=0
--fail-swap-on=false --anonymous-auth=false --protect-kernel-
defaults=true --client-ca-file=/etc/kubernetes/ssl/kube-
ca.pem --address=0.0.0.0 --cgroup-driver=cgroupfs --resolv-
conf=/run/systemd/resolve/resolv.conf
```

4.2.12 Verify that the RotateKubeletServerCertificate argument is set to true (Automated)

Result: Not Applicable

Remediation: Edit the kubelet service file /etc/systemd/system/kubelet.service.d/10-kubeadm.conf on each worker node and set the below parameter in KUBELET_CERTIFICATE_ARGS variable. --feature-gates=RotateKubeletServerCertificate=true Based on your system, restart the kubelet service. For example: systemctl daemon-reload systemctl restart kubelet.service

Clusters provisioned by RKE handles certificate rotation directly through RKE.

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

4.2.13 Ensure that the Kubelet only makes use of Strong Cryptographic Ciphers (Automated)

Result: pass

Remediation: 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_G
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
Based on your system, restart the kubelet service. For example: systemctl daemon-reload systemctl restart kubelet.service

Audit:

```
/bin/ps -fC kubelet
```

Audit Config:

```
/bin/cat /var/lib/kubelet/config.yaml
```

Expected Result:

```
'' is not present
```

5.1 RBAC and Service Accounts

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

Result: warn

Remediation: 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 Minimize access to secrets (Manual)

Result: warn

Remediation: Where possible, remove get, list and watch access to secret objects in the cluster.

5.1.3 Minimize wildcard use in Roles and ClusterRoles (Manual)

Result: warn

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

5.1.4 Minimize access to create pods (Manual)

Result: warn

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

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

Result: pass

Remediation: 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

Audit Script: `check_for_default_sa.sh`

```
#!/bin/bash

set -eE

handle_error() {
    echo "false"
}

trap 'handle_error' ERR

count_sa=$(kubectl get serviceaccounts --all-namespaces -o
json | jq -r '.items[] | select(.metadata.name=="default") |
select((.automountServiceAccountToken == null) or
(.automountServiceAccountToken == true))' | jq .metadata.names
pace | wc -l)
if [[ ${count_sa} -gt 0 ]]; then
    echo "false"
    exit
fi

for ns in $(kubectl get ns --no-headers -o custom-columns=":me
tadata.name")
do
    for result in $(kubectl get
clusterrolebinding,rolebinding -n $ns -o json | jq -r '.items[
] | select((.subjects[].kind=="ServiceAccount"
and .subjects[].name=="default") or
(.subjects[].kind=="Group"
and .subjects[].name=="system:serviceaccounts"))' | jq -r '"\
(.roleRef.kind),\(.roleRef.name)"')
    do
        read kind name <<<$(IFS=","; echo $result)
        resource_count=$(kubectl get $kind $name -n $ns -o
json | jq -r '.rules[] | select(.resources[] !=
"podsecuritypolicies")' | wc -l)
        if [[ ${resource_count} -gt 0 ]]; then
```

```
              echo "false"
              exit
          fi
      done
  done


  echo "true"
```

Audit Execution:

```
./check_for_default_sa.sh
```

Expected Result:

```
'true' is equal to 'true'
```

Returned Value:

```
true
```

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

Result: warn

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

5.2 Pod Security Policies

5.2.1 Minimize the admission of privileged containers (Manual)

Result: warn

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

5.2.2 Minimize the admission of containers wishing to share the host process ID namespace (Automated)

Result: pass

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

Audit:

```
kubectl get psp -o json | jq .items[] | jq -r 'select((.spec.h
ostPID == null) or (.spec.hostPID == false))' | jq .metadata.n
ame | wc -l | xargs -I {} echo '--count={}'
```

Expected Result:

```
1 is greater than 0
```

Returned Value:

```
--count=1
```

5.2.3 Minimize the admission of containers wishing to share the host IPC namespace (Automated)

Result: pass

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

Audit:

```
kubectl get psp -o json | jq .items[] | jq -r 'select((.spec.h
ostIPC == null) or (.spec.hostIPC == false))' | jq .metadata.n
ame | wc -l | xargs -I {} echo '--count={}'
```

Expected Result:

```
1 is greater than 0
```

Returned Value:

```
--count=1
```

5.2.4 Minimize the admission of containers wishing to share the host network namespace (Automated)

Result: pass

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

Audit:

```
kubectl get psp -o json | jq .items[] | jq -r 'select((.spec.h
ostNetwork == null) or (.spec.hostNetwork == false))' | jq .me
tadata.name | wc -l | xargs -I {} echo '--count={}'
```

Expected Result:

```
1 is greater than 0
```

Returned Value:

```
--count=1
```

5.2.5 Minimize the admission of containers with allowPrivilegeEscalation (Automated)

Result: pass

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

Audit:

```
kubectl get psp -o json | jq .items[] | jq -r 'select((.spec.a
llowPrivilegeEscalation == null) or
(.spec.allowPrivilegeEscalation == false))' | jq .metadata.nam
e | wc -l | xargs -I {} echo '--count={}'
```

Expected Result:

```
1 is greater than 0
```

Returned Value:

```
--count=1
```

5.2.6 Minimize the admission of root containers (Manual)

Result: warn

Remediation: 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.

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

Result: warn

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

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

Result: warn

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

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

Result: warn

Remediation: Review the use of capabilites in applications runnning on your cluster. Where a namespace contains applicaions which do not require any Linux capabilities to operate consider adding a PSP which forbids the admission of containers which do not drop all capabilities.

5.3 Network Policies and CNI

5.3.1 Ensure that the CNI in use supports Network Policies (Manual)

Result: warn

Remediation: 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 Ensure that all Namespaces have Network Policies defined (Automated)

Result: pass

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

Audit Script: `check_for_network_policies.sh`

```
#!/bin/bash

set -eE

handle_error() {
    echo "false"
}

trap 'handle_error' ERR

for namespace in $(kubectl get namespaces --all-namespaces -o
json | jq -r '.items[].metadata.name'); do
  policy_count=$(kubectl get networkpolicy -n ${namespace} -o
json | jq '.items | length')
  if [[ ${policy_count} -eq 0 ]]; then
    echo "false"
    exit
  fi
done

echo "true"
```

Audit Execution:

```
./check_for_network_policies.sh
```

Expected Result:

```
'true' is equal to 'true'
```

Returned Value:

```
true
```

5.4 Secrets Management

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

Result: warn

Remediation: if possible, rewrite application code to read secrets from mounted secret files, rather than from environment variables.

5.4.2 Consider external secret storage (Manual)

Result: warn

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

5.5 Extensible Admission Control

5.5.1 Configure Image Provenance using ImagePolicyWebhook admission controller (Manual)

Result: warn

Remediation: Follow the Kubernetes documentation and setup image provenance.

5.7 General Policies

5.7.1 Create administrative boundaries between resources using namespaces (Manual)

Result: warn

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

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

Result: warn

Remediation: Seccomp is an alpha feature currently. By default, all alpha features are disabled. So, you would need to enable alpha features in the apiserver by passing "--feature- gates=AllAlpha=true" argument. Edit the /etc/kubernetes/apiserver file on the master node and set the KUBE_API_ARGS parameter to "--feature-gates=AllAlpha=true" KUBE_API_ARGS="--feature-gates=AllAlpha=true" Based on your system, restart the kube-apiserver service. For example: systemctl restart kube-apiserver.service Use annotations to enable the docker/default seccomp profile in your pod definitions. An example is as below: apiVersion: v1 kind: Pod metadata: name: trustworthy-pod annotations: seccomp.security.alpha.kubernetes.io/pod: docker/default spec: containers: - name: trustworthy-container image: sotrustworthy:latest

5.7.3 Apply Security Context to Your Pods and Containers (Manual)

Result: warn

Remediation: 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.

5.7.4 The default namespace should not be used (Automated)

Result: pass

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

Audit Script: `check_for_default_ns.sh`

```
#!/bin/bash

set -eE

handle_error() {
    echo "false"
}

trap 'handle_error' ERR

count=$(kubectl get all -n default -o json | jq .items[] | jq
-r 'select((.metadata.name!="kubernetes"))' | jq .metadata.nam
e | wc -l)
if [[ ${count} -gt 0 ]]; then
    echo "false"
    exit
fi

echo "true"
```

Audit Execution:

```
./check_for_default_ns.sh
```

Expected Result:

```
'true' is equal to 'true'
```

Returned Value:

```
true
```