RKE2 CIS v1.23 Benchmark - Self-Assessment Guide - Rancher v2.6

Contents

CIS v1.23 Kubernetes Benchmark - Rancher v2.6 RKE2 with Kubernetes v1.22 up to v1.24

Click here to download a PDF version of this document.

Overview

This document is a companion to the Rancher v2.6 RKE2 security hardening guide. The hardening guide provides prescriptive guidance for hardening a production installation of Rancher with RKE2 provisioned clusters, 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 v1.23 Benchmark	Rancher v2.6.5+	CIS v1.23	Kubernetes v1.22 up to v1.24

Because Rancher and RKE2 install Kubernetes services as 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 v1.23. You can download the benchmark, after creating a free account, in Center for Internet Security (CIS).

Testing controls methodology

RKE2 launches control plane components as static pods, managed by the kubelet, and uses containerd as the container runtime.

Configuration is defined by arguments passed to the container at the time of initialization or via configuration file.

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 command line on the hosts of all RKE2 nodes. 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: Only `automated` tests (previously called `scored`) are covered in this guide.

Controls

1.1 Control Plane Node Configuration Files

1.1.1 Ensure that the API server pod specification 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 control plane node. For example, chmod 644 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml

Audit:

```
stat -c permissions=%a /var/lib/rancher/rke2/agent/pod-
manifests/kube-apiserver.yaml
```

Expected Result:

```
'permissions' is equal to '644'
```

Returned Value:

```
permissions=644
```

1.1.2 Ensure that the API server pod specification 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 control plane node. For example, chown root:root /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/pod-
manifests/kube-apiserver.yaml; then stat -c %U:%G /var/lib/
rancher/rke2/agent/pod-manifests/kube-apiserver.yaml; fi'
```

Expected Result:

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

Returned Value:

```
root:root
```

1.1.3 Ensure that the controller manager pod specification 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 control plane node. For example, chmod 644 /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml; then stat -c permissions=%a /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml; fi'
```

Expected Result:

```
'644' is equal to '644'
```

Returned Value:

```
permissions=644
```

1.1.4 Ensure that the controller manager pod specification 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 control plane node. For example, chown root:root /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml; then stat -c %U:%G /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml; fi'
```

Expected Result:

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

Returned Value:

```
root:root
```

1.1.5 Ensure that the scheduler pod specification 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 control plane node. For example, chmod 644 /var/lib/rancher/rke2/agent/pod-manifests/kube-scheduler.yaml

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/pod-
manifests/kube-scheduler.yaml; then stat -c permissions=%a /
var/lib/rancher/rke2/agent/pod-manifests/kube-scheduler.yaml;
fi'
```

Expected Result:

```
'644' is equal to '644'
```

Returned Value:

```
permissions=644
```

1.1.6 Ensure that the scheduler pod specification 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 control plane node. For example, chown root:root /var/lib/rancher/rke2/agent/pod-manifests/kube-scheduler.yaml

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/pod-
manifests/kube-scheduler.yaml; then stat -c %U:%G /var/lib/
rancher/rke2/agent/pod-manifests/kube-scheduler.yaml; fi'
```

Expected Result:

```
'root:root' is present
```

Returned Value:

```
root:root
```

1.1.7 Ensure that the etcd pod specification 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 control plane node. For example, chmod 644 /var/lib/rancher/rke2/agent/pod-manifests/etcd.yaml

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/pod-manifests/etcd.yaml; then find /var/lib/rancher/rke2/agent/pod-manifests/etcd.yaml -name '*etcd*' | xargs stat -c permissions=%a; fi'
```

Expected Result:

```
permissions has permissions 644, expected 644 or more restrictive
```

Returned Value:

```
permissions=644
```

1.1.8 Ensure that the etcd pod specification 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 control plane node. For example, chown root:root /var/lib/rancher/rke2/agent/pod-manifests/etcd.yaml

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/pod-manifests/etcd.yaml; then stat -c %U:%G /var/lib/rancher/rke2/agent/pod-manifests/etcd.yaml; fi'
```

Expected Result:

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

Returned Value:

```
root:root
```

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 control plane node. For example, chmod 644

Audit:

```
ps -ef | grep $kubeletbin | grep -- --cni-conf-dir | sed 's%.*
cni-conf-dir[= ]\([^ ]*\).*%\1%' | xargs -I{} find {} -
mindepth 1 | xargs --no-run-if-empty stat -c permissions=%a
find /var/lib/cni/networks -type f 2> /dev/null | xargs --no-
run-if-empty stat -c permissions=%a
```

Expected Result:

```
'permissions' is present
```

Returned Value:

```
Usage: grep [OPTION]... PATTERN [FILE]... Try 'grep --help'
for more information.
```

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 control plane node. For example, chown root:root

Audit:

```
ps -ef | grep $kubeletbin | grep -- --cni-conf-dir | sed 's%.*
cni-conf-dir[= ]\([^ ]*\).*%\1%' | xargs -I{} find {} -
mindepth 1 | xargs --no-run-if-empty stat -c %U:%G find /var/
```

```
lib/cni/networks -type f 2> /dev/null | xargs --no-run-if-
empty stat -c %U:%G
```

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 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 permissions=%a /var/lib/rancher/rke2/server/db/etcd
```

Expected Result:

```
permissions has permissions 700, expected 700 or more
restrictive
```

Returned Value:

```
permissions=700
```

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

Result: Not Applicable

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

1.1.13 Ensure that the admin.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 control plane node. For example, chmod 600 /etc/kubernetes/admin.conf

Audit:

```
stat -c permissions=%a /var/lib/rancher/rke2/server/cred/
admin.kubeconfig
```

Expected Result:

```
permissions has permissions 644, expected 644 or more
restrictive
```

Returned Value:

```
permissions=644
```

1.1.14 Ensure that the admin.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 control plane node. For example, chown root:root / etc/kubernetes/admin.conf

Audit:

```
stat -c %U:%G /var/lib/rancher/rke2/server/cred/
admin.kubeconfig
```

Expected Result:

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

Returned Value:

```
root:root
```

1.1.15 Ensure that the scheduler.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 control plane node. For example, chmod 644 scheduler

Audit:

```
stat -c %a /var/lib/rancher/rke2/server/cred/
scheduler.kubeconfig
```

Expected Result:

```
'644' is equal to '644'
```

Returned Value:

```
644
```

1.1.16 Ensure that the scheduler.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 control plane node. For example, chown root:root scheduler

Audit:

```
 stat -c %U:%G /var/lib/rancher/rke2/server/cred/
scheduler.kubeconfig
```

Expected Result:

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

Returned Value:

```
root:root
```

1.1.17 Ensure that the controller-manager.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 control plane node. For example, chmod 644 controllermanager

Audit:

```
 stat -c %a /var/lib/rancher/rke2/server/cred/
controller.kubeconfig
```

Expected Result:

```
'644' is equal to '644'
```

Returned Value:

```
644
```

1.1.18 Ensure that the controller-manager.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 control plane node. For example, chown root:root /var/lib/rancher/rke2/server/cred/controller.kubeconfig

Audit:

```
stat -c %U:%G /var/lib/rancher/rke2/server/cred/
controller.kubeconfig
```

Expected Result:

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

Returned Value:

```
root:root
```

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 control plane node. For example, chown -R root:root /etc/kubernetes/pki/

Audit:

```
stat -c %U:%G /var/lib/rancher/rke2/server/tls
```

Expected Result:

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

Returned Value:

```
root:root
```

1.1.20 Ensure that the Kubernetes PKI certificate 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 control plane node. For example, chmod -R 644 /var/lib/rancher/rke2/server/tls/*.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 /var/lib/rancher/rke2/server/tls/
*.crt
```

Expected Result:

```
'permissions' is present
```

Returned Value:

```
false
```

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

Result: warn

Remediation: Run the below command (based on the file location on your system) on the control plane node. For example, chmod -R 600 /var/lib/rancher/rke2/server/tls/*.key

Audit:

```
find /etc/kubernetes/pki/ -name '*.key' | xargs stat -c
permissions=%a
```

1.2 API Server

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

Result: warn

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and set the below parameter. --anonymous-auth=false

Audit:

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

1.2.2 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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
```

```
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
```

```
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

1.2.3 Ensure that the --DenyServiceExternalIPs is not set (Automated)

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and remove the `DenyServiceExternalIPs` from enabled admission plugins.

Audit:

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

Expected Result:

```
'--enable-admission-plugins' does not have
'DenyServiceExternalIPs' OR '--enable-admission-plugins' is
not present
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
```

```
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and remove the --kubelet-https parameter.

Audit:

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

Expected Result:

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

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
```

```
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
```

```
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
```

```
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
```

```
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and set the --kubelet-certificate-authority parameter to the path to the cert file for the certificate authority. --kubelet-certificate-authority=

Audit:

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

Expected Result:

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

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
```

```
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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:

```
'--authorization-mode' does not have 'AlwaysAllow'
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
```

```
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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:

```
'--authorization-mode' has 'Node'
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
```

```
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
```

```
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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:

```
'--authorization-mode' has 'RBAC'
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
```

```
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
```

```
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: warn

Remediation: Follow the Kubernetes documentation and set the desired limits in a configuration file. Then, edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-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:

```
'--enable-admission-plugins' has 'EventRateLimit'
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
```

```
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
```

```
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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:

```
'--enable-admission-plugins' does not have 'AlwaysAdmit' OR
'--enable-admission-plugins' is not present
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
```

```
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: warn

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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
```

Expected Result:

```
'--enable-admission-plugins' has 'AlwaysPullImages'
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
```

```
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
```

```
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true root 12676 1712 99 13:45 ? 00:00:00 kubectl
get --server=https://localhost:6443/ --client-certificate=/
var/lib/rancher/rke2/server/tls/client-kube-apiserver.crt --
client-key=/var/lib/rancher/rke2/server/tls/client-kube-
apiserver.key --certificate-authority=/var/lib/rancher/rke2/
server/tls/server-ca.crt --raw=/livez
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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
```

Expected Result:

```
'--enable-admission-plugins' has 'SecurityContextDeny' OR '--
enable-admission-plugins' has 'PodSecurityPolicy'
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
```

```
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
```

```
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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 present OR '--disable-
admission-plugins' is not present
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
```

```
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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 present OR '--disable-
admission-plugins' is not present
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
```

```
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
```

```
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

1.2.16 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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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:

```
'--enable-admission-plugins' has 'NodeRestriction'
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
```

```
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
```

```
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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:

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

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
```

```
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
```

```
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and set the below parameter. --profiling=false

Audit:

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

Expected Result:

```
'--profiling' is equal to 'false'
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
```

```
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
```

```
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: Not Applicable

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and set the --audit-log-path parameter to a suitable path and file where you would like audit logs to be written, for example, --audit-log-path=/var/log/apiserver/audit.log

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

Result: Not Applicable

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and set the --audit-log-maxage parameter to 30 or as an appropriate number of days, for example, --audit-log-maxage=30

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

Result: Not Applicable

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and set the --audit-log-maxbackup parameter to 10 or to an appropriate value. For example, --audit-log-maxbackup=10

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

Result: Not Applicable

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and set the --audit-log-maxsize parameter to an appropriate size in MB. For example, to set it as 100 MB, --audit-log-maxsize=100

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-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
present
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
```

```
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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 '--service-
account-lookup' is present
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
```

```
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
```

```
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and set the --service-account-key-file parameter to the public key file for service accounts. For example, --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 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
```

```
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
```

```
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

1.2.26 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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
```

```
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
```

```
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

1.2.27 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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
```

```
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

1.2.28 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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
```

```
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

1.2.29 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 /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane 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 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
```

```
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
```

```
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

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

Result: pass

Remediation: Follow the Kubernetes documentation and configure a EncryptionConfig file. Then, edit the API server pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-apiserver.yaml on the control plane node and set the --encryption-provider-config parameter to the path of that file. For example, --encryption-provider-config=

Audit:

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

Expected Result:

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

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
```

```
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
```

```
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

1.2.31 Ensure that encryption providers are appropriately configured (Manual)

Result: Not Applicable

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

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

Result: warn

Remediation: Edit the API server pod specification file /etc/kubernetes/manifests/kube-apiserver.yaml on the control plane node and set the below parameter. --tls-cipher-
suites=TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POL
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_128_GC
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_G
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305,TLS_ECDHE_ECDSA_WITH_CHACHA2
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_CBC_S
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_GCM_S
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256,TLS_RSA_WITH_3DES_EDE_CB
TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_

Audit:

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

Expected Result:

```
'--tls-cipher-suites' is present
```

Returned Value:

```
root 1772 1712 21 13:36 ? 00:01:56 kube-apiserver --advertise-
address=172.31.15.55 --allow-privileged=true --anonymous-
auth=false --api-audiences=https://
kubernetes.default.svc.cluster.local,rke2 --authorization-
mode=Node,RBAC --bind-address=0.0.0.0 --cert-dir=/var/lib/
rancher/rke2/server/tls/temporary-certs --client-ca-file=/var/
lib/rancher/rke2/server/tls/client-ca.crt --egress-selector-
config-file=/var/lib/rancher/rke2/server/etc/egress-selector-
config.yaml --enable-admission-
plugins=NodeRestriction,PodSecurityPolicy --enable-aggregator-
routing=true --encryption-provider-config=/var/lib/rancher/
rke2/server/cred/encryption-config.json --etcd-cafile=/var/
lib/rancher/rke2/server/tls/etcd/server-ca.crt --etcd-
certfile=/var/lib/rancher/rke2/server/tls/etcd/client.crt --
etcd-keyfile=/var/lib/rancher/rke2/server/tls/etcd/client.key
--etcd-servers=https://127.0.0.1:2379 --feature-
gates=JobTrackingWithFinalizers=true --kubelet-certificate-
authority=/var/lib/rancher/rke2/server/tls/server-ca.crt --
kubelet-client-certificate=/var/lib/rancher/rke2/server/tls/
client-kube-apiserver.crt --kubelet-client-key=/var/lib/
rancher/rke2/server/tls/client-kube-apiserver.key --kubelet-
preferred-address-types=InternalIP,ExternalIP,Hostname --
profiling=false --proxy-client-cert-file=/var/lib/rancher/
rke2/server/tls/client-auth-proxy.crt --proxy-client-key-
file=/var/lib/rancher/rke2/server/tls/client-auth-proxy.key --
requestheader-allowed-names=system:auth-proxy --requestheader-
client-ca-file=/var/lib/rancher/rke2/server/tls/request-
header-ca.crt --requestheader-extra-headers-prefix=X-Remote-
Extra- --requestheader-group-headers=X-Remote-Group --
requestheader-username-headers=X-Remote-User --secure-
```

```
port=6443 --service-account-issuer=https://
kubernetes.default.svc.cluster.local --service-account-key-
file=/var/lib/rancher/rke2/server/tls/service.key --service-
account-signing-key-file=/var/lib/rancher/rke2/server/tls/
service.key --service-cluster-ip-range=10.43.0.0/16 --service-
node-port-range=30000-32767 --storage-backend=etcd3 --tls-
cert-file=/var/lib/rancher/rke2/server/tls/serving-kube-
apiserver.crt --tls-private-key-file=/var/lib/rancher/rke2/
server/tls/serving-kube-apiserver.key root 1938 1828 2
13:36 ? 00:00:11 kube-controller-manager --flex-volume-plugin-
dir=/var/lib/kubelet/volumeplugins --terminated-pod-gc-
threshold=1000 --permit-port-sharing=true --allocate-node-
cidrs=true --authentication-kubeconfig=/var/lib/rancher/rke2/
server/cred/controller.kubeconfig --authorization-kubeconfig=/
var/lib/rancher/rke2/server/cred/controller.kubeconfig --bind-
address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/
kube-controller-manager --cluster-cidr=10.42.0.0/16 --cluster-
signing-kube-apiserver-client-cert-file=/var/lib/rancher/rke2/
server/tls/client-ca.crt --cluster-signing-kube-apiserver-
client-key-file=/var/lib/rancher/rke2/server/tls/client-
ca.key --cluster-signing-kubelet-client-cert-file=/var/lib/
rancher/rke2/server/tls/client-ca.crt --cluster-signing-
kubelet-client-key-file=/var/lib/rancher/rke2/server/tls/
client-ca.key --cluster-signing-kubelet-serving-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-kubelet-serving-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --cluster-signing-legacy-unknown-cert-file=/
var/lib/rancher/rke2/server/tls/server-ca.crt --cluster-
signing-legacy-unknown-key-file=/var/lib/rancher/rke2/server/
tls/server-ca.key --configure-cloud-routes=false --
controllers=*,-service,-route,-cloud-node-lifecycle --feature-
gates=JobTrackingWithFinalizers=true --kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
profiling=false --root-ca-file=/var/lib/rancher/rke2/server/
tls/server-ca.crt --secure-port=10257 --service-account-
private-key-file=/var/lib/rancher/rke2/server/tls/service.key
--service-cluster-ip-range=10.43.0.0/16 --use-service-account-
credentials=true
```

1.3 Controller Manager

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

Result: pass

Remediation: Edit the Controller Manager pod specification file /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml on the control plane node and set the --terminated-pod-gc-threshold to an appropriate threshold, for example, --terminated-pod-gc-threshold=10

Audit:

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

Expected Result:

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

Returned Value:

```
root 1938 1828 2 13:36 ? 00:00:11 kube-controller-manager --
flex-volume-plugin-dir=/var/lib/kubelet/volumeplugins --
terminated-pod-gc-threshold=1000 --permit-port-sharing=true --
allocate-node-cidrs=true --authentication-kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
authorization-kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --bind-address=127.0.0.1 --cert-dir=/
var/lib/rancher/rke2/server/tls/kube-controller-manager --
cluster-cidr=10.42.0.0/16 --cluster-signing-kube-apiserver-
client-cert-file=/var/lib/rancher/rke2/server/tls/client-
ca.crt --cluster-signing-kube-apiserver-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-client-cert-file=/var/lib/rancher/rke2/server/tls/
client-ca.crt --cluster-signing-kubelet-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-serving-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-kubelet-serving-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --cluster-signing-
```

```
legacy-unknown-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-legacy-unknown-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --configure-cloud-
routes=false --controllers=*,-service,-route,-cloud-node-
lifecycle --feature-gates=JobTrackingWithFinalizers=true --
kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --profiling=false --root-ca-file=/var/
lib/rancher/rke2/server/tls/server-ca.crt --secure-port=10257
--service-account-private-key-file=/var/lib/rancher/rke2/
server/tls/service.key --service-cluster-ip-
range=10.43.0.0/16 --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 /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml on the control plane node and set the below parameter. --profiling=false

Audit:

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

Expected Result:

```
'--profiling' is equal to 'false'
```

Returned Value:

```
root 1938 1828 2 13:36 ? 00:00:11 kube-controller-manager --
flex-volume-plugin-dir=/var/lib/kubelet/volumeplugins --
terminated-pod-gc-threshold=1000 --permit-port-sharing=true --
allocate-node-cidrs=true --authentication-kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
authorization-kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --bind-address=127.0.0.1 --cert-dir=/
var/lib/rancher/rke2/server/tls/kube-controller-manager --
cluster-cidr=10.42.0.0/16 --cluster-signing-kube-apiserver-
client-cert-file=/var/lib/rancher/rke2/server/tls/client-
ca.crt --cluster-signing-kube-apiserver-client-key-file=/var/
```

```
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-client-cert-file=/var/lib/rancher/rke2/server/tls/
client-ca.crt --cluster-signing-kubelet-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-serving-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-kubelet-serving-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --cluster-signing-
legacy-unknown-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-legacy-unknown-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --configure-cloud-
routes=false --controllers=*,-service,-route,-cloud-node-
lifecycle --feature-gates=JobTrackingWithFinalizers=true --
kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --profiling=false --root-ca-file=/var/
lib/rancher/rke2/server/tls/server-ca.crt --secure-port=10257
--service-account-private-key-file=/var/lib/rancher/rke2/
server/tls/service.key --service-cluster-ip-
range=10.43.0.0/16 --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 /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml on the control plane node to set the below parameter.
--use-service-account-credentials=true

Audit:

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

Expected Result:

```
'--use-service-account-credentials' is not equal to 'false'
```

Returned Value:

```
root 1938 1828 2 13:36 ? 00:00:11 kube-controller-manager --
flex-volume-plugin-dir=/var/lib/kubelet/volumeplugins --
terminated-pod-gc-threshold=1000 --permit-port-sharing=true --
allocate-node-cidrs=true --authentication-kubeconfig=/var/lib/
```

```
rancher/rke2/server/cred/controller.kubeconfig --
authorization-kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --bind-address=127.0.0.1 --cert-dir=/
var/lib/rancher/rke2/server/tls/kube-controller-manager --
cluster-cidr=10.42.0.0/16 --cluster-signing-kube-apiserver-
client-cert-file=/var/lib/rancher/rke2/server/tls/client-
ca.crt --cluster-signing-kube-apiserver-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-client-cert-file=/var/lib/rancher/rke2/server/tls/
client-ca.crt --cluster-signing-kubelet-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-serving-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-kubelet-serving-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --cluster-signing-
legacy-unknown-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-legacy-unknown-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --configure-cloud-
routes=false --controllers=*,-service,-route,-cloud-node-
lifecycle --feature-gates=JobTrackingWithFinalizers=true --
kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --profiling=false --root-ca-file=/var/
lib/rancher/rke2/server/tls/server-ca.crt --secure-port=10257
--service-account-private-key-file=/var/lib/rancher/rke2/
server/tls/service.key --service-cluster-ip-
range=10.43.0.0/16 --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 /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml on the control plane 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 1938 1828 2 13:36 ? 00:00:11 kube-controller-manager --
flex-volume-plugin-dir=/var/lib/kubelet/volumeplugins --
terminated-pod-gc-threshold=1000 --permit-port-sharing=true --
allocate-node-cidrs=true --authentication-kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
authorization-kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --bind-address=127.0.0.1 --cert-dir=/
var/lib/rancher/rke2/server/tls/kube-controller-manager --
cluster-cidr=10.42.0.0/16 --cluster-signing-kube-apiserver-
client-cert-file=/var/lib/rancher/rke2/server/tls/client-
ca.crt --cluster-signing-kube-apiserver-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-client-cert-file=/var/lib/rancher/rke2/server/tls/
client-ca.crt --cluster-signing-kubelet-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-serving-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-kubelet-serving-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --cluster-signing-
legacy-unknown-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-legacy-unknown-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --configure-cloud-
routes=false --controllers=*,-service,-route,-cloud-node-
lifecycle --feature-gates=JobTrackingWithFinalizers=true --
kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --profiling=false --root-ca-file=/var/
lib/rancher/rke2/server/tls/server-ca.crt --secure-port=10257
--service-account-private-key-file=/var/lib/rancher/rke2/
server/tls/service.key --service-cluster-ip-
range=10.43.0.0/16 --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 /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml on the control plane 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 1938 1828 2 13:36 ? 00:00:11 kube-controller-manager --
flex-volume-plugin-dir=/var/lib/kubelet/volumeplugins --
terminated-pod-gc-threshold=1000 --permit-port-sharing=true --
allocate-node-cidrs=true --authentication-kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
authorization-kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --bind-address=127.0.0.1 --cert-dir=/
var/lib/rancher/rke2/server/tls/kube-controller-manager --
cluster-cidr=10.42.0.0/16 --cluster-signing-kube-apiserver-
client-cert-file=/var/lib/rancher/rke2/server/tls/client-
ca.crt --cluster-signing-kube-apiserver-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-client-cert-file=/var/lib/rancher/rke2/server/tls/
client-ca.crt --cluster-signing-kubelet-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-serving-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-kubelet-serving-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --cluster-signing-
legacy-unknown-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-legacy-unknown-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --configure-cloud-
routes=false --controllers=*,-service,-route,-cloud-node-
lifecycle --feature-gates=JobTrackingWithFinalizers=true --
kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --profiling=false --root-ca-file=/var/
lib/rancher/rke2/server/tls/server-ca.crt --secure-port=10257
--service-account-private-key-file=/var/lib/rancher/rke2/
```

```
server/tls/service.key --service-cluster-ip-
range=10.43.0.0/16 --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 /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml on the control plane node and set the --feature-gates parameter to include RotateKubeletServerCertificate=true. --feature-gates=RotateKubeletServerCertificate=true

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 /var/lib/rancher/rke2/agent/pod-manifests/kube-controller-manager.yaml on the control plane 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 equal to '127.0.0.1' OR '--bind-address'
is not present
```

Returned Value:

```
root 1938 1828 2 13:36 ? 00:00:11 kube-controller-manager --
flex-volume-plugin-dir=/var/lib/kubelet/volumeplugins --
terminated-pod-gc-threshold=1000 --permit-port-sharing=true --
allocate-node-cidrs=true --authentication-kubeconfig=/var/lib/
rancher/rke2/server/cred/controller.kubeconfig --
authorization-kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --bind-address=127.0.0.1 --cert-dir=/
var/lib/rancher/rke2/server/tls/kube-controller-manager --
cluster-cidr=10.42.0.0/16 --cluster-signing-kube-apiserver-
client-cert-file=/var/lib/rancher/rke2/server/tls/client-
```

```
ca.crt --cluster-signing-kube-apiserver-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-client-cert-file=/var/lib/rancher/rke2/server/tls/
client-ca.crt --cluster-signing-kubelet-client-key-file=/var/
lib/rancher/rke2/server/tls/client-ca.key --cluster-signing-
kubelet-serving-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-kubelet-serving-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --cluster-signing-
legacy-unknown-cert-file=/var/lib/rancher/rke2/server/tls/
server-ca.crt --cluster-signing-legacy-unknown-key-file=/var/
lib/rancher/rke2/server/tls/server-ca.key --configure-cloud-
routes=false --controllers=*,-service,-route,-cloud-node-
lifecycle --feature-gates=JobTrackingWithFinalizers=true --
kubeconfig=/var/lib/rancher/rke2/server/cred/
controller.kubeconfig --profiling=false --root-ca-file=/var/
lib/rancher/rke2/server/tls/server-ca.crt --secure-port=10257
--service-account-private-key-file=/var/lib/rancher/rke2/
server/tls/service.key --service-cluster-ip-
range=10.43.0.0/16 --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 /var/lib/rancher/rke2/agent/pod-manifests/kube-scheduler.yaml file on the control plane node and set the below parameter. --profiling=false

Audit:

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

Expected Result:

```
'--profiling' is equal to 'false'
```

Returned Value:

```
root 1949 1809 0 13:36 ? 00:00:03 kube-scheduler --permit-port-sharing=true --authentication-kubeconfig=/var/lib/rancher/rke2/server/cred/scheduler.kubeconfig --authorization-kubeconfig=/var/lib/rancher/rke2/server/cred/scheduler.kubeconfig --bind-address=127.0.0.1 --cert-dir=/var/lib/rancher/rke2/server/tls/kube-scheduler --kubeconfig=/var/lib/rancher/rke2/server/cred/scheduler.kubeconfig --profiling=false --secure-port=10259
```

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 /var/lib/rancher/rke2/agent/pod-manifests/kube-scheduler.yaml on the control plane 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 equal to '127.0.0.1' OR '--bind-address'
is not present
```

Returned Value:

```
root 1949 1809 0 13:36 ? 00:00:03 kube-scheduler --permit-
port-sharing=true --authentication-kubeconfig=/var/lib/
rancher/rke2/server/cred/scheduler.kubeconfig --authorization-
kubeconfig=/var/lib/rancher/rke2/server/cred/
scheduler.kubeconfig --bind-address=127.0.0.1 --cert-dir=/var/
lib/rancher/rke2/server/tls/kube-scheduler --kubeconfig=/var/
lib/rancher/rke2/server/cred/scheduler.kubeconfig --
profiling=false --secure-port=10259
```

2 Etcd Node Configuration

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

Result: Not Applicable

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=

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

Result: Not Applicable

Remediation: Edit the etcd pod specification file /var/lib/rancher/rke2/agent/pod-manifests/etcd.yaml on the master node and set the below parameter. --client-cert-auth="true"

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

Result: pass

Remediation: Edit the etcd pod specification file /var/lib/rancher/rke2/agent/pod-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:

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

Returned Value:

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin HOSTNAME=rke2-test-cis-pool2-a43ee26f-wkkxx container=oci
ETCD_UNSUPPORTED_ARCH=
FILE_HASH=d2bdd17bc97578f8271f4d4a03819fc5a46cd240011eb0daafc2
419a22747787 NO_PROXY=.svc,.cluster.local,
10.42.0.0/16,10.43.0.0/16 HOME=/root
```

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

Result: Not Applicable

Remediation: Follow the etcd service documentation and configure peer TLS encryption as appropriate for your etcd cluster. Then, edit the etcd pod specification file /var/lib/rancher/rke2/agent/pod-manifests/etcd.yaml on the master node and set the below parameters. --peer-client-file= --peer-key-file=

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

Result: Not Applicable

Remediation: Edit the etcd pod specification file /var/lib/rancher/rke2/agent/pod-manifests/etcd.yaml on the master node and set the below parameter. --peer-client-cert-auth=true

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

Result: pass

Remediation: Edit the etcd pod specification file /var/lib/rancher/rke2/agent/pod-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:

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

Returned Value:

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin HOSTNAME=rke2-test-cis-pool2-a43ee26f-wkkxx container=oci
```

```
ETCD_UNSUPPORTED_ARCH=
FILE_HASH=d2bdd17bc97578f8271f4d4a03819fc5a46cd240011eb0daafc2
419a22747787 NO_PROXY=.svc,.cluster.local,
10.42.0.0/16,10.43.0.0/16 HOME=/root
```

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

Result: warn

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 /var/lib/rancher/rke2/agent/pod-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:

```
'ETCD_TRUSTED_CA_FILE' is present
```

Returned Value:

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin HOSTNAME=rke2-test-cis-pool2-a43ee26f-wkkxx container=oci
ETCD_UNSUPPORTED_ARCH=
FILE_HASH=d2bdd17bc97578f8271f4d4a03819fc5a46cd240011eb0daafc2
419a22747787 NO_PROXY=.svc,.cluster.local,
10.42.0.0/16,10.43.0.0/16 HOME=/root
```

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: Not Applicable

Remediation: Create an audit policy file for your cluster.

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

Result: warn

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

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: Run the below command (based on the file location on your system) on the each worker node. For example, chmod 644 /etc/systemd/system/kubelet.service.d/10-kubeadm.conf

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

Result: Not Applicable

Remediation: Run the below command (based on the file location on your system) on the each worker node. For example, chown root:root /etc/systemd/system/kubelet.service.d/10-kubeadm.conf

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

Result: pass

Remediation: Run the below command (based on the file location on your system) on the each worker node. For example, chmod 644 /var/lib/rancher/rke2/agent/kubeproxy.kubeconfig

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/
kubeproxy.kubeconfig; then stat -c permissions=%a /var/lib/
rancher/rke2/agent/kubeproxy.kubeconfig; fi'
```

Expected Result:

```
permissions has permissions 644, expected 644 or more
restrictive OR '/var/lib/rancher/rke2/agent/
kubeproxy.kubeconfig' is not present
```

Returned Value:

```
permissions=644 permissions=644 permissions=644
```

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

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 /var/lib/rancher/rke2/agent/kubeproxy.kubeconfig

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/kubeproxy.kubeconfig; then stat -c %U:%G /var/lib/rancher/rke2/agent/kubeproxy.kubeconfig; fi'
```

Expected Result:

```
'root:root' is present OR '/var/lib/rancher/rke2/agent/kubeproxy.kubeconfig' is not present
```

Returned Value:

```
root:root root:root root:root
```

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 /var/lib/rancher/rke2/agent/kubelet.kubeconfig

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/kubelet.kubeconfig; then stat -c permissions=%a /var/lib/rancher/rke2/agent/kubelet.kubeconfig; fi'
```

Expected Result:

```
'644' is equal to '644'
```

Returned Value:

```
permissions=644 permissions=644 permissions=644
```

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 /var/lib/rancher/rke2/agent/kubelet.kubeconfig

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/
kubelet.kubeconfig; then stat -c %U:%G /var/lib/rancher/rke2/
agent/kubelet.kubeconfig; fi'
```

Expected Result:

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

Returned Value:

```
root:root root:root root:root
```

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

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 $1}')
CAFILE=/node$CAFILE
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 has permissions 600, expected 644 or more
restrictive
```

Returned Value:

```
permissions=600 permissions=600 permissions=600
```

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

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 $1}')
CAFILE=/node$CAFILE
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 equal to 'root:root'
```

Returned Value:

```
root:root root:root root:root
```

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

Result: pass

Remediation: Run the following command (using the config file location identified in the Audit step) chmod 644 /var/lib/rancher/rke2/agent/kubelet.kubeconfig

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/kubelet.kubeconfig; then stat -c permissions=%a /var/lib/rancher/rke2/agent/kubelet.kubeconfig; fi'
```

Expected Result:

```
'644' is equal to '644'
```

Returned Value:

```
permissions=644 permissions=644 permissions=644
```

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

Result: pass

Remediation: Run the following command (using the config file location identified in the Audit step) chown root:root /var/lib/rancher/rke2/agent/kubelet.kubeconfig

Audit:

```
/bin/sh -c 'if test -e /var/lib/rancher/rke2/agent/kubelet.kubeconfig; then stat -c %U:%G /var/lib/rancher/rke2/agent/kubelet.kubeconfig; fi'
```

Expected Result:

```
'root:root' is present
```

Returned Value:

```
root:root root:root root:root
```

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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'--anonymous-auth' is equal to 'false'
```

Returned Value:

```
UID PID PPID C STIME TTY TIME CMD root 1627 1603 1 13:39 ?
00:00:07 kubelet --volume-plugin-dir=/var/lib/kubelet/
volumeplugins --file-check-frequency=5s --sync-frequency=30s
--address=0.0.0.0 --alsologtostderr=false --anonymous-
auth=false --authentication-token-webhook=true --
authorization-mode=Webhook --cgroup-driver=systemd --client-
ca-file=/var/lib/rancher/rke2/agent/client-ca.crt --cloud-
provider=external --cluster-dns=10.43.0.10 --cluster-
domain=cluster.local --container-runtime-endpoint=unix:///run/
k3s/containerd/containerd.sock --containerd=/run/k3s/
containerd/containerd.sock --eviction-
```

```
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool1-611e63c8-g6fc5 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/machine=0dbad31b-8d5d-4265-9c5a-
b1ff1e9eec31 --pod-infra-container-image=index.docker.io/
rancher/pause:3.6 --pod-manifest-path=/var/lib/rancher/rke2/
agent/pod-manifests --read-only-port=0 --resolv-conf=/run/
systemd/resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1600 1552 2 13:35 ? 00:00:12 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool2-a43ee26f-wkkxx --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=660f91e5-00bf-4e4a-90ee-0c39b5622b96 --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/etcd:NoExecute --resolv-conf=/run/systemd/
```

```
resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1612 1583 3 13:36 ? 00:00:16 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool3-9a073952-blrq8 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=fa9bfc7f-9194-47b6-8adc-4bd2150acf9a --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/control-plane:NoSchedule --resolv-conf=/
run/systemd/resolve/resolv.conf --serialize-image-pulls=false
--stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key
```

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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'--authorization-mode' does not have 'AlwaysAllow'
```

Returned Value:

```
UID PID PPID C STIME TTY TIME CMD root 1627 1603 1 13:39 ?
00:00:07 kubelet --volume-plugin-dir=/var/lib/kubelet/
volumeplugins --file-check-frequency=5s --sync-frequency=30s
--address=0.0.0.0 --alsologtostderr=false --anonymous-
auth=false --authentication-token-webhook=true --
authorization-mode=Webhook --cgroup-driver=systemd --client-
ca-file=/var/lib/rancher/rke2/agent/client-ca.crt --cloud-
provider=external --cluster-dns=10.43.0.10 --cluster-
domain=cluster.local --container-runtime-endpoint=unix:///run/
k3s/containerd/containerd.sock --containerd=/run/k3s/
containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool1-611e63c8-g6fc5 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/machine=0dbad31b-8d5d-4265-9c5a-
b1ff1e9eec31 --pod-infra-container-image=index.docker.io/
rancher/pause:3.6 --pod-manifest-path=/var/lib/rancher/rke2/
agent/pod-manifests --read-only-port=0 --resolv-conf=/run/
systemd/resolve/resolv.conf --serialize-image-pulls=false --
```

```
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1600 1552 2 13:35 ? 00:00:12 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool2-a43ee26f-wkkxx --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=660f91e5-00bf-4e4a-90ee-0c39b5622b96 --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/etcd:NoExecute --resolv-conf=/run/systemd/
resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1612 1583 3 13:36 ? 00:00:16 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
```

```
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool3-9a073952-blrq8 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=fa9bfc7f-9194-47b6-8adc-4bd2150acf9a --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/control-plane:NoSchedule --resolv-conf=/
run/systemd/resolve/resolv.conf --serialize-image-pulls=false
--stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key
```

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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

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

Returned Value:

```
UID PID PPID C STIME TTY TIME CMD root 1627 1603 1 13:39 ?
00:00:07 kubelet --volume-plugin-dir=/var/lib/kubelet/
volumeplugins --file-check-frequency=5s --sync-frequency=30s
--address=0.0.0.0 --alsologtostderr=false --anonymous-
auth=false --authentication-token-webhook=true --
authorization-mode=Webhook --cgroup-driver=systemd --client-
ca-file=/var/lib/rancher/rke2/agent/client-ca.crt --cloud-
provider=external --cluster-dns=10.43.0.10 --cluster-
domain=cluster.local --container-runtime-endpoint=unix:///run/
k3s/containerd/containerd.sock --containerd=/run/k3s/
containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool1-611e63c8-g6fc5 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/machine=0dbad31b-8d5d-4265-9c5a-
b1ff1e9eec31 --pod-infra-container-image=index.docker.io/
rancher/pause:3.6 --pod-manifest-path=/var/lib/rancher/rke2/
agent/pod-manifests --read-only-port=0 --resolv-conf=/run/
systemd/resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1600 1552 2 13:35 ? 00:00:12 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
```

```
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool2-a43ee26f-wkkxx --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=660f91e5-00bf-4e4a-90ee-0c39b5622b96 --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/etcd:NoExecute --resolv-conf=/run/systemd/
resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1612 1583 3 13:36 ? 00:00:16 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool3-9a073952-blrq8 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=fa9bfc7f-9194-47b6-8adc-4bd2150acf9a --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
```

```
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/control-plane:NoSchedule --resolv-conf=/
run/systemd/resolve/resolv.conf --serialize-image-pulls=false
--stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key
```

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

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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'--read-only-port' is equal to '0' OR '--read-only-port' is
not present
```

Returned Value:

```
UID PID PPID C STIME TTY TIME CMD root 1627 1603 1 13:39 ?
00:00:07 kubelet --volume-plugin-dir=/var/lib/kubelet/
volumeplugins --file-check-frequency=5s --sync-frequency=30s
--address=0.0.0.0 --alsologtostderr=false --anonymous-
auth=false --authentication-token-webhook=true --
authorization-mode=Webhook --cgroup-driver=systemd --client-
ca-file=/var/lib/rancher/rke2/agent/client-ca.crt --cloud-
provider=external --cluster-dns=10.43.0.10 --cluster-
```

```
domain=cluster.local --container-runtime-endpoint=unix:///run/
k3s/containerd/containerd.sock --containerd=/run/k3s/
containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool1-611e63c8-g6fc5 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/machine=0dbad31b-8d5d-4265-9c5a-
b1ff1e9eec31 --pod-infra-container-image=index.docker.io/
rancher/pause:3.6 --pod-manifest-path=/var/lib/rancher/rke2/
agent/pod-manifests --read-only-port=0 --resolv-conf=/run/
systemd/resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1600 1552 2 13:35 ? 00:00:12 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool2-a43ee26f-wkkxx --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=660f91e5-00bf-4e4a-90ee-0c39b5622b96 --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
```

```
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/etcd:NoExecute --resolv-conf=/run/systemd/
resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1612 1583 3 13:36 ? 00:00:16 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool3-9a073952-blrq8 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=fa9bfc7f-9194-47b6-8adc-4bd2150acf9a --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/control-plane:NoSchedule --resolv-conf=/
run/systemd/resolve/resolv.conf --serialize-image-pulls=false
--stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key
```

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

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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'{.streamingConnectionIdleTimeout}' is present OR
'{.streamingConnectionIdleTimeout}' is not present
```

Returned Value:

```
apiVersion: v1 clusters: - cluster: server: https://
127.0.0.1:6443 certificate-authority: /var/lib/rancher/rke2/
agent/server-ca.crt name: local contexts: - context: cluster:
local namespace: default user: user name: Default current-
context: Default kind: Config preferences: {} users: - name:
user user: client-certificate: /var/lib/rancher/rke2/agent/
client-kubelet.crt client-key: /var/lib/rancher/rke2/agent/
client-kubelet.key apiVersion: v1 clusters: - cluster:
server: https://127.0.0.1:6443 certificate-authority: /var/
lib/rancher/rke2/agent/server-ca.crt name: local contexts: -
context: cluster: local namespace: default user: user name:
Default current-context: Default kind: Config preferences: {}
users: - name: user user: client-certificate: /var/lib/
rancher/rke2/agent/client-kubelet.crt client-key: /var/lib/
rancher/rke2/agent/client-kubelet.key apiVersion: v1
clusters: - cluster: server: https://127.0.0.1:6443
certificate-authority: /var/lib/rancher/rke2/agent/server-
ca.crt name: local contexts: - context: cluster: local
namespace: default user: user name: Default current-context:
```

```
Default kind: Config preferences: {} users: - name: user user: client-certificate: /var/lib/rancher/rke2/agent/client-kubelet.crt client-key: /var/lib/rancher/rke2/agent/client-kubelet.key
```

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` to `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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'{.protectKernelDefaults}' is present OR '{.protectKernelDefaults}' is not present
```

Returned Value:

```
apiVersion: v1 clusters: - cluster: server: https://127.0.0.1:6443 certificate-authority: /var/lib/rancher/rke2/agent/server-ca.crt name: local contexts: - context: cluster: local namespace: default user: user name: Default current-context: Default kind: Config preferences: {} users: - name: user user: client-certificate: /var/lib/rancher/rke2/agent/client-kubelet.crt client-key: /var/lib/rancher/rke2/agent/client-kubelet.key apiVersion: v1 clusters: - cluster: server: https://127.0.0.1:6443 certificate-authority: /var/lib/rancher/rke2/agent/server-ca.crt name: local contexts: - context: cluster: local namespace: default user: user name:
```

```
Default current-context: Default kind: Config preferences: {}
users: - name: user user: client-certificate: /var/lib/
rancher/rke2/agent/client-kubelet.crt client-key: /var/lib/
rancher/rke2/agent/client-kubelet.key apiVersion: v1
clusters: - cluster: server: https://127.0.0.1:6443
certificate-authority: /var/lib/rancher/rke2/agent/server-
ca.crt name: local contexts: - context: cluster: local
namespace: default user: user name: Default current-context:
Default kind: Config preferences: {} users: - name: user
user: client-certificate: /var/lib/rancher/rke2/agent/client-
kubelet.crt client-key: /var/lib/rancher/rke2/agent/client-
kubelet.key
```

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` to `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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'{.makeIPTablesUtilChains}' is present OR
'{.makeIPTablesUtilChains}' is not present
```

Returned Value:

```
apiVersion: v1 clusters: - cluster: server: https://
127.0.0.1:6443 certificate-authority: /var/lib/rancher/rke2/
agent/server-ca.crt name: local contexts: - context: cluster:
```

```
local namespace: default user: user name: Default current-
context: Default kind: Config preferences: {} users: - name:
user user: client-certificate: /var/lib/rancher/rke2/agent/
client-kubelet.crt client-key: /var/lib/rancher/rke2/agent/
client-kubelet.key apiVersion: v1 clusters: - cluster:
server: https://127.0.0.1:6443 certificate-authority: /var/
lib/rancher/rke2/agent/server-ca.crt name: local contexts: -
context: cluster: local namespace: default user: user name:
Default current-context: Default kind: Config preferences: {}
users: - name: user user: client-certificate: /var/lib/
rancher/rke2/agent/client-kubelet.crt client-key: /var/lib/
rancher/rke2/agent/client-kubelet.key apiVersion: v1
clusters: - cluster: server: https://127.0.0.1:6443
certificate-authority: /var/lib/rancher/rke2/agent/server-
ca.crt name: local contexts: - context: cluster: local
namespace: default user: user name: Default current-context:
Default kind: Config preferences: {} users: - name: user
user: client-certificate: /var/lib/rancher/rke2/agent/client-
kubelet.crt client-key: /var/lib/rancher/rke2/agent/client-
kubelet.key
```

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

Result: warn

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

Audit:

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

Expected Result:

```
'--hostname-override' is not present
```

Returned Value:

```
UID PID PPID C STIME TTY TIME CMD root 1627 1603 1 13:39 ?
00:00:07 kubelet --volume-plugin-dir=/var/lib/kubelet/
volumeplugins --file-check-frequency=5s --sync-frequency=30s
--address=0.0.0.0 --alsologtostderr=false --anonymous-
auth=false --authentication-token-webhook=true --
authorization-mode=Webhook --cgroup-driver=systemd --client-
ca-file=/var/lib/rancher/rke2/agent/client-ca.crt --cloud-
provider=external --cluster-dns=10.43.0.10 --cluster-
domain=cluster.local --container-runtime-endpoint=unix:///run/
k3s/containerd/containerd.sock --containerd=/run/k3s/
containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool1-611e63c8-g6fc5 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/machine=0dbad31b-8d5d-4265-9c5a-
b1ff1e9eec31 --pod-infra-container-image=index.docker.io/
rancher/pause:3.6 --pod-manifest-path=/var/lib/rancher/rke2/
agent/pod-manifests --read-only-port=0 --resolv-conf=/run/
systemd/resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1600 1552 2 13:35 ? 00:00:12 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
```

```
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool2-a43ee26f-wkkxx --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=660f91e5-00bf-4e4a-90ee-0c39b5622b96 --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/etcd:NoExecute --resolv-conf=/run/systemd/
resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1612 1583 3 13:36 ? 00:00:16 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool3-9a073952-blrq8 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=fa9bfc7f-9194-47b6-8adc-4bd2150acf9a --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/control-plane:NoSchedule --resolv-conf=/
```

```
run/systemd/resolve/resolv.conf --serialize-image-pulls=false
--stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key
```

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

Result: warn

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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'{.eventRecordQPS}' is present
```

Returned Value:

```
apiVersion: v1 clusters: - cluster: server: https://
127.0.0.1:6443 certificate-authority: /var/lib/rancher/rke2/
agent/server-ca.crt name: local contexts: - context: cluster:
local namespace: default user: user name: Default current-
context: Default kind: Config preferences: {} users: - name:
user user: client-certificate: /var/lib/rancher/rke2/agent/
client-kubelet.crt client-key: /var/lib/rancher/rke2/agent/
client-kubelet.key apiVersion: v1 clusters: - cluster:
server: https://127.0.0.1:6443 certificate-authority: /var/
lib/rancher/rke2/agent/server-ca.crt name: local contexts: -
context: cluster: local namespace: default user: user name:
Default current-context: Default kind: Config preferences: {}
```

```
users: - name: user user: client-certificate: /var/lib/
rancher/rke2/agent/client-kubelet.crt client-key: /var/lib/
rancher/rke2/agent/client-kubelet.key apiVersion: v1
clusters: - cluster: server: https://127.0.0.1:6443
certificate-authority: /var/lib/rancher/rke2/agent/server-
ca.crt name: local contexts: - context: cluster: local
namespace: default user: user name: Default current-context:
Default kind: Config preferences: {} users: - name: user
user: client-certificate: /var/lib/rancher/rke2/agent/client-
kubelet.crt client-key: /var/lib/rancher/rke2/agent/client-
kubelet.key
```

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

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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

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

Returned Value:

```
UID PID PPID C STIME TTY TIME CMD root 1627 1603 1 13:39 ?
00:00:07 kubelet --volume-plugin-dir=/var/lib/kubelet/
```

```
volumeplugins --file-check-frequency=5s --sync-frequency=30s
--address=0.0.0.0 --alsologtostderr=false --anonymous-
auth=false --authentication-token-webhook=true --
authorization-mode=Webhook --cgroup-driver=systemd --client-
ca-file=/var/lib/rancher/rke2/agent/client-ca.crt --cloud-
provider=external --cluster-dns=10.43.0.10 --cluster-
domain=cluster.local --container-runtime-endpoint=unix:///run/
k3s/containerd/containerd.sock --containerd=/run/k3s/
containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool1-611e63c8-g6fc5 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/machine=0dbad31b-8d5d-4265-9c5a-
b1ff1e9eec31 --pod-infra-container-image=index.docker.io/
rancher/pause:3.6 --pod-manifest-path=/var/lib/rancher/rke2/
agent/pod-manifests --read-only-port=0 --resolv-conf=/run/
systemd/resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1600 1552 2 13:35 ? 00:00:12 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool2-a43ee26f-wkkxx --
```

```
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=660f91e5-00bf-4e4a-90ee-0c39b5622b96 --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/etcd:NoExecute --resolv-conf=/run/systemd/
resolve/resolv.conf --serialize-image-pulls=false --
stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key UID PID PPID C STIME
TTY TIME CMD root 1612 1583 3 13:36 ? 00:00:16 kubelet --
volume-plugin-dir=/var/lib/kubelet/volumeplugins --file-check-
frequency=5s --sync-frequency=30s --address=0.0.0.0 --
alsologtostderr=false --anonymous-auth=false --authentication-
token-webhook=true --authorization-mode=Webhook --cgroup-
driver=systemd --client-ca-file=/var/lib/rancher/rke2/agent/
client-ca.crt --cloud-provider=external --cluster-
dns=10.43.0.10 --cluster-domain=cluster.local --container-
runtime-endpoint=unix:///run/k3s/containerd/containerd.sock --
containerd=/run/k3s/containerd/containerd.sock --eviction-
hard=imagefs.available<5%,nodefs.available<5% --eviction-
minimum-reclaim=imagefs.available=10%,nodefs.available=10% --
fail-swap-on=false --healthz-bind-address=127.0.0.1 --
hostname-override=rke2-test-cis-pool3-9a073952-blrq8 --
kubeconfig=/var/lib/rancher/rke2/agent/kubelet.kubeconfig --
log-file=/var/lib/rancher/rke2/agent/logs/kubelet.log --log-
file-max-size=50 --logtostderr=false --node-
labels=rke.cattle.io/
machine=fa9bfc7f-9194-47b6-8adc-4bd2150acf9a --pod-infra-
container-image=index.docker.io/rancher/pause:3.6 --pod-
manifest-path=/var/lib/rancher/rke2/agent/pod-manifests --
read-only-port=0 --register-with-taints=node-
role.kubernetes.io/control-plane:NoSchedule --resolv-conf=/
run/systemd/resolve/resolv.conf --serialize-image-pulls=false
--stderrthreshold=FATAL --tls-cert-file=/var/lib/rancher/rke2/
```

```
agent/serving-kubelet.crt --tls-private-key-file=/var/lib/
rancher/rke2/agent/serving-kubelet.key
```

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` to `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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'{.rotateCertificates}' is present OR '{.rotateCertificates}'
is not present
```

Returned Value:

```
apiVersion: v1 clusters: - cluster: server: https://
127.0.0.1:6443 certificate-authority: /var/lib/rancher/rke2/
agent/server-ca.crt name: local contexts: - context: cluster:
local namespace: default user: user name: Default current-
context: Default kind: Config preferences: {} users: - name:
user user: client-certificate: /var/lib/rancher/rke2/agent/
client-kubelet.crt client-key: /var/lib/rancher/rke2/agent/
client-kubelet.key apiVersion: v1 clusters: - cluster:
server: https://127.0.0.1:6443 certificate-authority: /var/
lib/rancher/rke2/agent/server-ca.crt name: local contexts: -
context: cluster: local namespace: default user: user name:
Default current-context: Default kind: Config preferences: {}
```

```
users: - name: user user: client-certificate: /var/lib/
rancher/rke2/agent/client-kubelet.crt client-key: /var/lib/
rancher/rke2/agent/client-kubelet.key apiVersion: v1
clusters: - cluster: server: https://127.0.0.1:6443
certificate-authority: /var/lib/rancher/rke2/agent/server-
ca.crt name: local contexts: - context: cluster: local
namespace: default user: user name: Default current-context:
Default kind: Config preferences: {} users: - name: user
user: client-certificate: /var/lib/rancher/rke2/agent/client-
kubelet.crt client-key: /var/lib/rancher/rke2/agent/client-
kubelet.key
```

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

Result: pass

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

Audit:

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

Audit Config:

```
/bin/cat /var/lib/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'{.featureGates.RotateKubeletServerCertificate}' is present
OR '{.featureGates.RotateKubeletServerCertificate}' is not
present
```

Returned Value:

```
apiVersion: v1 clusters: - cluster: server: https://
127.0.0.1:6443 certificate-authority: /var/lib/rancher/rke2/
agent/server-ca.crt name: local contexts: - context: cluster:
local namespace: default user: user name: Default current-
```

```
context: Default kind: Config preferences: {} users: - name:
user user: client-certificate: /var/lib/rancher/rke2/agent/
client-kubelet.crt client-key: /var/lib/rancher/rke2/agent/
client-kubelet.key apiVersion: v1 clusters: - cluster:
server: https://127.0.0.1:6443 certificate-authority: /var/
lib/rancher/rke2/agent/server-ca.crt name: local contexts: -
context: cluster: local namespace: default user: user name:
Default current-context: Default kind: Config preferences: {}
users: - name: user user: client-certificate: /var/lib/
rancher/rke2/agent/client-kubelet.crt client-key: /var/lib/
rancher/rke2/agent/client-kubelet.key apiVersion: v1
clusters: - cluster: server: https://127.0.0.1:6443
certificate-authority: /var/lib/rancher/rke2/agent/server-
ca.crt name: local contexts: - context: cluster: local
namespace: default user: user name: Default current-context:
Default kind: Config preferences: {} users: - name: user
user: client-certificate: /var/lib/rancher/rke2/agent/client-
kubelet.crt client-key: /var/lib/rancher/rke2/agent/client-
kubelet.key
```

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

Result: warn

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/rancher/rke2/agent/kubelet.kubeconfig
```

Expected Result:

```
'{range .tlsCipherSuites[:]}{}{','}{end}' is present
```

Returned Value:

```
apiVersion: v1 clusters: - cluster: server: https://
127.0.0.1:6443 certificate-authority: /var/lib/rancher/rke2/
agent/server-ca.crt name: local contexts: - context: cluster:
local namespace: default user: user name: Default current-
context: Default kind: Config preferences: {} users: - name:
user user: client-certificate: /var/lib/rancher/rke2/agent/
client-kubelet.crt client-key: /var/lib/rancher/rke2/agent/
client-kubelet.key apiVersion: v1 clusters: - cluster:
server: https://127.0.0.1:6443 certificate-authority: /var/
lib/rancher/rke2/agent/server-ca.crt name: local contexts: -
context: cluster: local namespace: default user: user name:
Default current-context: Default kind: Config preferences: {}
users: - name: user user: client-certificate: /var/lib/
rancher/rke2/agent/client-kubelet.crt client-key: /var/lib/
rancher/rke2/agent/client-kubelet.key apiVersion: v1
clusters: - cluster: server: https://127.0.0.1:6443
certificate-authority: /var/lib/rancher/rke2/agent/server-
ca.crt name: local contexts: - context: cluster: local
namespace: default user: user name: Default current-context:
Default kind: Config preferences: {} users: - name: user
user: client-certificate: /var/lib/rancher/rke2/agent/client-
kubelet.crt client-key: /var/lib/rancher/rke2/agent/client-
kubelet.key
```

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. (Manual)

Result: warn

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

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.

5.1.7 Avoid use of system:masters group (Manual)

Result: warn

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

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

Result: warn

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

5.2 Pod Security Standards

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

Result: warn

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

5.2.2 Minimize the admission of privileged containers (Manual)

Result: warn

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

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

Result: warn

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

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

Result: warn

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

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

Result: warn

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

5.2.6 Minimize the admission of containers with allowPrivilegeEscalation (Automated)

Result: warn

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

5.2.7 Minimize the admission of root containers (Automated)

Result: warn

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

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

Result: warn

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

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

Result: warn

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

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

Result: warn

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

5.2.11 Minimize the admission of Windows HostProcess containers (Manual)

Result: warn

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

5.2.12 Minimize the admission of HostPath volumes (Manual)

Result: warn

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

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

Result: warn

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

5.3 Network Policies and CNI

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

Result: warn

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

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

5.7.3 Apply SecurityContext to your Pods and Containers (Manual)

Result: warn

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

5.7.4 The default namespace should not be used (Manual)

Result: warn

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.