# CIS Benchmark Rancher Self-Assessment Guide - v2.4

# Contents

# CIS Kubernetes Benchmark v1.5 - Rancher v2.4 with Kubernetes v1.15

Click here to download a PDF version of this document

## Overview

This document is a companion to the Rancher v2.4 security hardening guide. The hardening guide provides prescriptive guidance for hardening a production installation of Rancher, and this benchmark guide is meant to help you evaluate the level of security of the hardened cluster against each control in the benchmark.

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

| Self Assessment Guide Version | Rancher Version | Hardening Guide Version | Kubernetes Version | CIS Benchmark Version |
|---|---|---|---|---|
| Self Assessment Guide v2.4 | Rancher v2.4 | Hardening Guide v2.4 | Kubernetes v1.15 | Benchmark v1.5 |

Because Rancher and RKE install Kubernetes services as Docker containers, many of the control verification checks in the CIS Kubernetes Benchmark don't 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 v1.5. You can download the benchmark after logging in to **CISecurity.org**.

## Testing controls methodology

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

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

> NOTE: only scored tests are covered in this guide.

## Controls

---

# 1 Master Node Security Configuration

## 1.1 Master Node Configuration Files

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

**Result:** Not Applicable

**Remediation:** RKE doesn't require or maintain a configuration file for the API server. All configuration is passed in as arguments at container run time.

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

**Result:** Not Applicable

**Remediation:** RKE doesn't require or maintain a configuration file for the API server. All configuration is passed in as arguments at container run time.

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

**Result:** Not Applicable

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

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

**Result:** Not Applicable

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

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

**Result:** Not Applicable

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

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

**Result:** Not Applicable

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

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

**Result:** Not Applicable

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

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

**Result:** Not Applicable

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

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

**Result:** PASS

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

```
ps -ef | grep etcd
```

Run the below command (based on the etcd data directory found above). For example,

```
chmod 700 /var/lib/etcd
```

**Audit Script:** 1.1.11.sh

```
#!/bin/bash -e

etcd_bin=${1}

test_dir=$(ps -ef | grep ${etcd_bin} | grep -- --data-dir |
sed 's%.*data-dir[= ]\([^ ]*\).*%\1%')

docker inspect etcd | jq -r '.[].HostConfig.Binds[]' | grep "$
{test_dir}" | cut -d ":" -f 1 | xargs stat -c %a
```

**Audit Execution:**

```
./1.1.11.sh etcd
```

**Expected result**:

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

## 1.1.12 Ensure that the etcd data directory ownership is set to `etcd:etcd` (Scored)

**Result:** PASS

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

```
ps -ef | grep etcd
```

Run the below command (based on the etcd data directory found above). For example,

```
chown etcd:etcd /var/lib/etcd
```

**Audit Script:** 1.1.12.sh

```
#!/bin/bash -e

etcd_bin=${1}

test_dir=$(ps -ef | grep ${etcd_bin} | grep -- --data-dir |
sed 's%.*data-dir[= ]\([^ ]*\).*%\1%')

docker inspect etcd | jq -r '.[].HostConfig.Binds[]' | grep "$
{test_dir}" | cut -d ":" -f 1 | xargs stat -c %U:%G
```

**Audit Execution:**

```
./1.1.12.sh etcd
```

**Expected result**:

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

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

**Result:** Not Applicable

**Remediation:** RKE does not store the kubernetes default kubeconfig credentials file on the nodes. It's presented to user where RKE is run. We recommend that this `kube_config_cluster.yml` file be kept in secure store.

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

**Result:** Not Applicable

**Remediation:** RKE does not store the kubernetes default kubeconfig credentials file on the nodes. It's presented to user where RKE is run. We recommend that this `kube_config_cluster.yml` file be kept in secure store.

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

**Result:** Not Applicable

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

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

**Result:** Not Applicable

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

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

**Result:** Not Applicable

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

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

**Result:** Not Applicable

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

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

**Result:** PASS

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

```
chown -R root:root /etc/kubernetes/ssl
```

**Audit:**

```
stat -c %U:%G /etc/kubernetes/ssl
```

**Expected result**:

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

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

**Result:** PASS

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

```
chmod -R 644 /etc/kubernetes/ssl
```

**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 '/etc/kubernetes/ssl/*.pem'
```

**Expected result**:

```
'true' is present
```

## 1.1.21 Ensure that the Kubernetes PKI key file permissions are set to `600` (Scored)

**Result:** PASS

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

```
chmod -R 600 /etc/kubernetes/ssl/certs/serverca
```

**Audit Script:** 1.1.21.sh

```
#!/bin/bash -e
check_dir=${1:-/etc/kubernetes/ssl}

for file in $(find ${check_dir} -name "*key.pem"); do
    file_permission=$(stat -c %a ${file})
  if [[ "${file_permission}" == "600" ]]; then
    continue
  else
    echo "FAIL: ${file} ${file_permission}"
    exit 1
  fi
done

echo "pass"
```

**Audit Execution:**

```
./1.1.21.sh /etc/kubernetes/ssl
```

**Expected result**:

```
'pass' is present
```

## 1.2 API Server

### 1.2.2 Ensure that the `--basic-auth-file` argument is not set (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result**:

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

## 1.2.3 Ensure that the `--token-auth-file` parameter is not set (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result**:

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

## 1.2.4 Ensure that the `--kubelet-https` argument is set to true (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

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

```
--kubelet-client-certificate=<path/to/client-certificate-file>
--kubelet-client-key=<path/to/client-key-file>
```

**Audit:**

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

**Expected result**:

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

## 1.2.6 Ensure that the `--kubelet-certificate-authority` argument is set as appropriate (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result**:

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

### 1.2.7 Ensure that the `--authorization-mode` argument is not set to `AlwaysAllow` (Scored)

**Result:** PASS

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

```
--authorization-mode=RBAC
```

**Audit:**

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

**Expected result**:

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

### 1.2.8 Ensure that the `--authorization-mode` argument includes `Node` (Scored)

**Result:** PASS

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

```
--authorization-mode=Node,RBAC
```

**Audit:**

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

**Expected result**:

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

## 1.2.9 Ensure that the `--authorization-mode` argument includes `RBAC` (Scored)

**Result:** PASS

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

```
--authorization-mode=Node,RBAC
```

**Audit:**

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

**Expected result**:

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

## 1.2.11 Ensure that the admission control plugin `AlwaysAdmit` is not set (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result**:

```
'NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Priority,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecurityPolicy,EventRateLimit' not have 'AlwaysAdmit' OR '--enable-admission-plugins' is not present
```

### 1.2.14 Ensure that the admission control plugin `ServiceAccount` is set (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result**:

```
'NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Priority,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecurityPolicy,EventRateLimit' has 'ServiceAccount' OR '--enable-admission-plugins' is not present
```

### 1.2.15 Ensure that the admission control plugin `NamespaceLifecycle` is set (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result**:

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

## 1.2.16 Ensure that the admission control plugin `PodSecurityPolicy` is set (Scored)

**Result:** PASS

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

```
  --enable-admission-plugins=...,PodSecurityPolicy,...
```

Then restart the API Server.

**Audit:**

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

**Expected result**:

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

## 1.2.17 Ensure that the admission control plugin `NodeRestriction` is set (Scored)

**Result:** PASS

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

```
--enable-admission-plugins=...,NodeRestriction,...
```

**Audit:**

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

**Expected result:**

```
'NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,ValidatingAdmissionWebhook,ResourceQuota,NodeRestriction,Priority,TaintNodesByCondition,PersistentVolumeClaimResize,PodSecurityPolicy,EventRateLimit' has 'NodeRestriction'
```

## 1.2.18 Ensure that the `--insecure-bind-address` argument is not set (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result:**

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

## 1.2.19 Ensure that the `--insecure-port` argument is set to `0` (Scored)

**Result:** PASS

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

```
--insecure-port=0
```

**Audit:**

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

**Expected result:**

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

## 1.2.20 Ensure that the `--secure-port` argument is not set to `0` (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result:**

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

## 1.2.21 Ensure that the `--profiling` argument is set to `false` (Scored)

**Result:** PASS

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

```
--profiling=false
```

**Audit:**

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

**Expected result:**

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

### 1.2.22 Ensure that the `--audit-log-path` argument is set (Scored)

**Result:** PASS

**Remediation:** Edit the API server pod specification file `/etc/kubernetes/manifests/kube-apiserver.yaml` on the master node and set the `--audit-log-path` parameter to a suitable path and file where you would like audit logs to be written, for example:

```
--audit-log-path=/var/log/apiserver/audit.log
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

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

```
--audit-log-maxage=30
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

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

```
--audit-log-maxbackup=10
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

**Remediation:** Edit the API server pod specification file `/etc/kubernetes/manifests/kube-apiserver.yaml` on the master node and set the `--audit-log-maxsize` parameter to an appropriate size in **MB**. For example, to set it as `100` **MB**:

```
--audit-log-maxsize=100
```

**Audit:**

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

**Expected result**:

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

## 1.2.26 Ensure that the `--request-timeout` argument is set as appropriate (Scored)

**Result:** PASS

**Remediation:** Edit the API server pod specification file `/etc/kubernetes/manifests/kube-apiserver.yaml` and set the below parameter as appropriate and if needed. For example,

```
--request-timeout=300s
```

**Audit:**

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

**Expected result**:

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

## 1.2.27 Ensure that the `--service-account-lookup` argument is set to `true` (Scored)

**Result:** PASS

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

```
--service-account-lookup=true
```

Alternatively, you can delete the `--service-account-lookup` parameter from this file so that the default takes effect.

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

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

```
--service-account-key-file=<filename>
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

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

```
--etcd-certfile=<path/to/client-certificate-file>
--etcd-keyfile=<path/to/client-key-file>
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

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

```
--tls-cert-file=<path/to/tls-certificate-file>
--tls-private-key-file=<path/to/tls-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
```

## 1.2.31 Ensure that the `--client-ca-file` argument is set as appropriate (Scored)

**Result:** PASS

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

```
--client-ca-file=<path/to/client-ca-file>
```

**Audit:**

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

**Expected result**:

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

## 1.2.32 Ensure that the `--etcd-cafile` argument is set as appropriate (Scored)

**Result:** PASS

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

```
--etcd-cafile=<path/to/ca-file>
```

**Audit:**

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

**Expected result**:

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

## 1.2.33 Ensure that the `--encryption-provider-config` argument is set as appropriate (Scored)

**Result:** PASS

**Remediation:** Follow the Kubernetes documentation and configure a EncryptionConfig file. Then, edit the API server pod specification file `/etc/kubernetes/manifests/kube-apiserver.yaml` on the master node and set the `--encryption-provider-config` parameter to the path of that file:

```
--encryption-provider-config=</path/to/EncryptionConfig/File>
```

**Audit:**

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

**Expected result**:

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

### 1.2.34 Ensure that encryption providers are appropriately configured (Scored)

**Result:** PASS

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

**Audit Script:** 1.2.34.sh

```
#!/bin/bash -e

check_file=${1}

grep -q -E 'aescbc|kms|secretbox' ${check_file}
if [ $? -eq 0 ]; then
  echo "--pass"
  exit 0
else
  echo "fail: encryption provider found in ${check_file}"
  exit 1
fi
```

**Audit Execution:**

```
./1.2.34.sh /etc/kubernetes/ssl/encryption.yaml
```

**Expected result**:

```
'--pass' is present
```

## 1.3 Controller Manager

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

**Result:** PASS

**Remediation:** Edit the Controller Manager pod specification file `/etc/kubernetes/manifests/kube-controller-manager.yaml` on the master

node and set the `--terminated-pod-gc-threshold` to an appropriate threshold, for example:

```
--terminated-pod-gc-threshold=10
```

**Audit:**

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

**Expected result**:

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

## 1.3.2 Ensure that the `--profiling` argument is set to false (Scored)

**Result:** PASS

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

```
--profiling=false
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

**Remediation:** Edit the Controller Manager pod specification file `/etc/kubernetes/manifests/kube-controller-manager.yaml` on the master node to set the below parameter.

```
--use-service-account-credentials=true
```

**Audit:**

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

**Expected result:**

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

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

**Result:** PASS

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

```
--service-account-private-key-file=<filename>
```

**Audit:**

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

**Expected result:**

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

## 1.3.5 Ensure that the `--root-ca-file` argument is set as appropriate (Scored)

**Result:** PASS

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

```
--root-ca-file=<path/to/file>
```

**Audit:**

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

**Expected result**:

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

## 1.3.6 Ensure that the `RotateKubeletServerCertificate` argument is set to `true` (Scored)

**Result:** PASS

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

```
--feature-gates=RotateKubeletServerCertificate=true
```

**Audit:**

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

**Expected result**:

```
'RotateKubeletServerCertificate=true' is equal to
'RotateKubeletServerCertificate=true'
```

## 1.3.7 Ensure that the `--bind-address argument` is set to `127.0.0.1` (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result**:

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

## 1.4 Scheduler

### 1.4.1 Ensure that the `--profiling` argument is set to `false` (Scored)

**Result:** PASS

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

```
--profiling=false
```

**Audit:**

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

**Expected result**:

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

### 1.4.2 Ensure that the `--bind-address` argument is set to `127.0.0.1` (Scored)

**Result:** PASS

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

**Audit:**

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

**Expected result**:

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

# 2 Etcd Node Configuration

## 2 Etcd Node Configuration Files

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

**Result:** PASS

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

```
--cert-file=</path/to/ca-file>
--key-file=</path/to/key-file>
```

**Audit:**

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

**Expected result:**

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

### 2.2 Ensure that the `--client-cert-auth` argument is set to `true` (Scored)

**Result:** PASS

**Remediation:** Edit the etcd pod specification file `/etc/kubernetes/manifests/etcd.yaml` on the master node and set the below parameter.

```
--client-cert-auth="true"
```

**Audit:**

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

**Expected result**:

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

## 2.3 Ensure that the `--auto-tls` argument is not set to `true` (Scored)

**Result:** PASS

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

```
    --auto-tls=false
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

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

```
    --peer-client-file=</path/to/peer-cert-file>
    --peer-key-file=</path/to/peer-key-file>
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

**Remediation:** Edit the etcd pod specification file `/etc/kubernetes/manifests/etcd.yaml` on the master node and set the below parameter.

```
--peer-client-cert-auth=true
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

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

```
--peer-auto-tls=false
```

**Audit:**

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

**Expected result**:

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

# 3 Control Plane Configuration

## 3.2 Logging

### 3.2.1 Ensure that a minimal audit policy is created (Scored)

**Result:** PASS

**Remediation:** Create an audit policy file for your cluster.

**Audit Script:** 3.2.1.sh

```
#!/bin/bash -e

api_server_bin=${1}

/bin/ps -ef | /bin/grep ${api_server_bin} | /bin/grep -v ${0}
| /bin/grep -v grep
```

**Audit Execution:**

```
./3.2.1.sh kube-apiserver
```

**Expected result**:

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

# 4 Worker Node Security Configuration

## 4.1 Worker Node Configuration Files

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

**Result:** Not Applicable

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

### 4.1.2 Ensure that the kubelet service file ownership is set to `root:root` (Scored)

**Result:** Not Applicable

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

### 4.1.3 Ensure that the proxy kubeconfig file permissions are set to `644` or more restrictive (Scored)

**Result:** PASS

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

```
chmod 644 /etc/kubernetes/ssl/kubecfg-kube-proxy.yaml
```

**Audit:**

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

**Expected result**:

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

## 4.1.4 Ensure that the proxy kubeconfig file ownership is set to `root:root` (Scored)

**Result:** PASS

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

```
chown root:root /etc/kubernetes/ssl/kubecfg-kube-proxy.yaml
```

**Audit:**

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

**Expected result**:

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

## 4.1.5 Ensure that the kubelet.conf file permissions are set to `644` or more restrictive (Scored)

**Result:** PASS

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

```
chmod 644 /etc/kubernetes/ssl/kubecfg-kube-node.yaml
```

**Audit:**

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

**Expected result**:

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

### 4.1.6 Ensure that the kubelet.conf file ownership is set to `root:root` (Scored)

**Result:** PASS

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

```
chown root:root /etc/kubernetes/ssl/kubecfg-kube-node.yaml
```

**Audit:**

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

**Expected result**:

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

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

**Result:** PASS

**Remediation:** Run the following command to modify the file permissions of the

```
--client-ca-file chmod 644 <filename>
```

**Audit:**

```
stat -c %a /etc/kubernetes/ssl/kube-ca.pem
```

**Expected result**:

```
'644' is equal to '644' OR '640' is present OR '600' is
present
```

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

**Result:** PASS

**Remediation:** Run the following command to modify the ownership of the `--client-ca-file`.

```
chown root:root <filename>
```

**Audit:**

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

**Expected result**:

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

### 4.1.9 Ensure that the kubelet configuration file has permissions set to `644` or more restrictive (Scored)

**Result:** Not Applicable

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

### 4.1.10 Ensure that the kubelet configuration file ownership is set to `root:root` (Scored)

**Result:** Not Applicable

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

## 4.2 Kubelet

### 4.2.1 Ensure that the `--anonymous-auth argument` is set to false (Scored)

**Result:** PASS

**Remediation:** If using a Kubelet config file, edit the file to set authentication: `anonymous`: enabled to `false`. If using executable arguments, edit the kubelet service file `/etc/systemd/system/kubelet.service.d/10-kubeadm.conf` on each worker node and set the below parameter in `KUBELET_SYSTEM_PODS_ARGS` variable.

```
--anonymous-auth=false
```

Based on your system, restart the kubelet service. For example:

```
systemctl daemon-reload
systemctl restart kubelet.service
```

**Audit:**

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

**Audit Config:**

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

**Expected result**:

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

### 4.2.2 Ensure that the `--authorization-mode` argument is not set to `AlwaysAllow` (Scored)

**Result:** PASS

**Remediation:** If using a Kubelet config file, edit the file to set authorization: `mode` to `Webhook`. If using executable arguments, edit the kubelet service file `/etc/systemd/system/kubelet.service.d/10-kubeadm.conf` on each worker node and set the below parameter in `KUBELET_AUTHZ_ARGS` variable.

```
--authorization-mode=Webhook
```

Based on your system, restart the kubelet service. For example:

```
systemctl daemon-reload
systemctl restart kubelet.service
```

**Audit:**

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

**Audit Config:**

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

**Expected result**:

```
'Webhook' not have 'AlwaysAllow'
```

### 4.2.3 Ensure that the `--client-ca-file` argument is set as appropriate (Scored)

**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=<path/to/client-ca-file>
```

Based on your system, restart the kubelet service. For example:

```
systemctl daemon-reload
systemctl restart kubelet.service
```

**Audit:**

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

**Audit Config:**

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

**Expected result**:

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

### 4.2.4 Ensure that the `--read-only-port` argument is set to `0` (Scored)

**Result:** PASS

**Remediation:** If using a Kubelet config file, edit the file to set `readOnlyPort` to `0`. If using command line arguments, edit the kubelet service file `/etc/systemd/system/kubelet.service.d/10-kubeadm.conf` on each worker node and set the below parameter in `KUBELET_SYSTEM_PODS_ARGS` variable.

```
--read-only-port=0
```

Based on your system, restart the kubelet service. For example:

```
systemctl daemon-reload
systemctl restart kubelet.service
```

**Audit:**

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

**Audit Config:**

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

**Expected result**:

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

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

**Result:** PASS

**Remediation:** If using a Kubelet config file, edit the file to set `streamingConnectionIdleTimeout` to a value other than `0`. If using command line arguments, edit the kubelet service file `/etc/systemd/`

`system/kubelet.service.d/10-kubeadm.conf` on each worker node and set the below parameter in `KUBELET_SYSTEM_PODS_ARGS` variable.

```
--streaming-connection-idle-timeout=5m
```

Based on your system, restart the kubelet service. For example:

```
systemctl daemon-reload
systemctl restart kubelet.service
```

**Audit:**

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

**Audit Config:**

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

**Expected result**:

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

### 4.2.6 Ensure that the `--protect-kernel-defaults` argument is set to `true` (Scored)

**Result:** PASS

**Remediation:** If using a Kubelet config file, edit the file to set `protectKernelDefaults`: `true`. If using command line arguments, edit the kubelet service file `/etc/systemd/system/kubelet.service.d/10-kubeadm.conf` on each worker node and set the below parameter in `KUBELET_SYSTEM_PODS_ARGS` variable.

```
--protect-kernel-defaults=true
```

Based on your system, restart the kubelet service. For example:

```
systemctl daemon-reload
systemctl restart kubelet.service
```

**Audit:**

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

**Audit Config:**

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

**Expected result**:

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

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

**Result:** PASS

**Remediation:** If using a Kubelet config file, edit the file to set `makeIPTablesUtilChains`: `true`. If using command line arguments, edit the kubelet service file `/etc/systemd/system/kubelet.service.d/10-kubeadm.conf` on each worker node and remove the `--make-iptables-util-chains` argument from the `KUBELET_SYSTEM_PODS_ARGS` variable. Based on your system, restart the kubelet service. For example:

```
systemctl daemon-reload
systemctl restart kubelet.service
```

**Audit:**

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

**Audit Config:**

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

**Expected result**:

```
'true' is equal to 'true' OR '--make-iptables-util-chains' is not present
```

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

**Result:** Not Applicable

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

### 4.2.11 Ensure that the `--rotate-certificates` argument is not set to `false` (Scored)

**Result:** PASS

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

```
systemctl daemon-reload
systemctl restart kubelet.service
```

**Audit:**

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

**Audit Config:**

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

**Expected result:**

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

### 4.2.12 Ensure that the `RotateKubeletServerCertificate` argument is set to `true` (Scored)

**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/kubelet/config.yaml
```

**Expected result**:

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

# 5 Kubernetes Policies

## 5.1 RBAC and Service Accounts

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

**Result:** PASS

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

```
automountServiceAccountToken: false
```

**Audit Script:** 5.1.5.sh

```
#!/bin/bash

export KUBECONFIG=${KUBECONFIG:-/root/.kube/config}

kubectl version > /dev/null
if [ $? -ne 0 ]; then
  echo "fail: kubectl failed"
  exit 1
fi

accounts="$(kubectl --kubeconfig=${KUBECONFIG} get
serviceaccounts -A -o json | jq -r '.items[] |
select(.metadata.name=="default") |
select((.automountServiceAccountToken == null) or
(.automountServiceAccountToken == true)) | "fail \
(.metadata.name) \(.metadata.namespace)"')"

if [[ "${accounts}" != "" ]]; then
  echo "fail: automountServiceAccountToken not false for
accounts: ${accounts}"
  exit 1
```

```
fi

default_binding="$(kubectl get
rolebindings,clusterrolebindings -A -o json | jq -r '.items[]
| select(.subjects[].kind=="ServiceAccount"
and .subjects[].name=="default"
and .metadata.name=="default").metadata.uid' | wc -l)"

if [[ "${default_binding}" -gt 0 ]]; then
    echo "fail: default service accounts have non default
bindings"
    exit 1
fi

echo "--pass"
exit 0
```

**Audit Execution:**

```
./5.1.5.sh
```

**Expected result:**

```
'--pass' is present
```

## 5.2 Pod Security Policies

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

**Result:** PASS

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

**Audit:**

```
kubectl --kubeconfig=/root/.kube/config get psp -o json |
jq .items[] | jq -r 'select((.spec.hostPID == null) or
```

```
(.spec.hostPID == false))' | jq .metadata.name | wc -l |
xargs -I {} echo '--count={}'
```

**Expected result**:

```
1 is greater than 0
```

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

**Result:** PASS

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

**Audit:**

```
kubectl --kubeconfig=/root/.kube/config get psp -o json |
jq .items[] | jq -r 'select((.spec.hostIPC == null) or
(.spec.hostIPC == false))' | jq .metadata.name | wc -l |
xargs -I {} echo '--count={}'
```

**Expected result**:

```
1 is greater than 0
```

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

**Result:** PASS

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

**Audit:**

```
kubectl --kubeconfig=/root/.kube/config get psp -o json |
jq .items[] | jq -r 'select((.spec.hostNetwork == null) or
(.spec.hostNetwork == false))' | jq .metadata.name | wc -l |
xargs -I {} echo '--count={}'
```

**Expected result**:

```
1 is greater than 0
```

## 5.2.5 Minimize the admission of containers with `allowPrivilegeEscalation` (Scored)

**Result:** PASS

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

**Audit:**

```
kubectl --kubeconfig=/root/.kube/config get psp -o json |
jq .items[] | jq -r 'select((.spec.allowPrivilegeEscalation
== null) or (.spec.allowPrivilegeEscalation == false))' |
jq .metadata.name | wc -l | xargs -I {} echo '--count={}'
```

**Expected result**:

```
1 is greater than 0
```

# 5.3 Network Policies and CNI

## 5.3.2 Ensure that all Namespaces have Network Policies defined (Scored)

**Result:** PASS

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

**Audit Script:** 5.3.2.sh

```
#!/bin/bash -e

export KUBECONFIG=${KUBECONFIG:-"/root/.kube/config"}

kubectl version > /dev/null
if [ $? -ne 0 ]; then
```

```
  echo "fail: kubectl failed"
  exit 1
fi

for namespace in $(kubectl get namespaces -A -o json | jq -r
'.items[].metadata.name'); do
  policy_count=$(kubectl get networkpolicy -n ${namespace} -o
json | jq '.items | length')
  if [ ${policy_count} -eq 0 ]; then
    echo "fail: ${namespace}"
    exit 1
  fi
done

echo "pass"
```

**Audit Execution:**

```
./5.3.2.sh
```

**Expected result**:

```
'pass' is present
```

## 5.6 General Policies

### 5.6.4 The default namespace should not be used (Scored)

**Result:** PASS

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

**Audit Script:** 5.6.4.sh

```
#!/bin/bash -e

export KUBECONFIG=${KUBECONFIG:-/root/.kube/config}

kubectl version > /dev/null
```

```
if [[ $? -gt 0 ]]; then
  echo "fail: kubectl failed"
  exit 1
fi

default_resources=$(kubectl get all -o json | jq --compact-
output '.items[] | select((.kind == "Service") and
(.metadata.name == "kubernetes") and (.metadata.namespace ==
"default") | not)' | wc -l)

echo "--count=${default_resources}"
```

**Audit Execution:**

```
./5.6.4.sh
```

**Expected result:**

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