# Rancher CIS Kubernetes v1.3.0 Benchmark Self Assessment

**Rancher v2.1.x**

*Version 1.0.0 - Nov 2018*

## Authors

Jason Greathouse

# Overview

The following document scores an RKE cluster provisioned according to the Rancher 2.1.x hardening guide against the CIS 1.3.0 Kubernetes benchmark. This document is to be used by Rancher operators, security teams, auditors and decision makers.

### 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.

Scoring the commands is different in Rancher Labs than in the CIS Benchmark. Where the commands differ from the original CIS benchmark, the 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` command to provide human-readable formatting.

Tests will have an exit code of zero on success and non-zero on failure.

### Known Scored Control Failures

The following scored controls do not currently pass, and Rancher Labs is working towards addressing these through future enhancements to the product.

- 1.1.21 - Ensure that the `--kubelet-certificate-authority` argument is set as appropriate (Scored)
- 1.3.6 - Ensure that the `RotateKubeletServerCertificate` argument is set to `true` (Scored)
- 1.4.11 - Ensure that the etcd data directory permissions are set to `700` or more-restrictive (Scored)
- 1.4.12 - Ensure that the etcd data directory ownership is set to `etcd:etcd` (Scored)
- 2.1.3 - Ensure that the `--authorization-mode` argument is not set to `AlwaysAllow` (Scored)
- 2.1.9 - Ensure that the `--hostname-override` argument is not set (Scored)
- 2.1.13 - Ensure that the `--rotate-certificates` argument is not set to `false` (Scored)
- 2.1.14 - Ensure that the `RotateKubeletServerCertificate` argument is set to `true` (Scored)

# Scored Results

# 1 - Master Node Security Configuration

## 1.1 - API Server

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--anonymous-auth=false").string'
```

**Returned Value:** `--anonymous-auth=false`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--basic-auth-file=.*").string'
```

**Returned Value:** `null`

**Result:** Pass

### 1.1.3 - Ensure that the `--insecure-allow-any-token` argument is not set (Scored)

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--insecure-allow-any-token").string'
```

**Returned Value:** `null`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--kubelet-https=false").string'
```

**Returned Value:** `null`

**Result:** Pass

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

**Notes**

Flag not set or `--insecure-bind-address=127.0.0.1`. RKE sets this flag to `--insecure-bind-address=127.0.0.1`

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--insecure-
bind-address=(?:(?!127\\.0\\.0\\.1).)+")'
```

**Returned Value:** `null`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--insecure-
port=0").string'
```

**Returned Value:** `--insecure-port=0`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--secure-
port=6443").string'
```

**Returned Value:** `--secure-port=6443`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--profiling=false").string'
```

**Returned Value:** `--profiling=false`

**Result:** Pass

### 1.1.9 - Ensure that the `--repair-malformed-updates` argument is set to `false` (Scored)

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--repair-malformed-updates=false").string'
```

**Returned Value:** `--repair-malformed-updates=false`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--enable-admission-plugins=.*(AlwaysAdmit).*").captures[].string'
```

**Returned Value:** `null`

**Result:** Pass

### 1.1.11 - Ensure that the admission control plugin `AlwaysPullImages` is set (Scored)

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--enable-admission-plugins=.*(AlwaysPullImages).*").captures[].string'
```

**Returned Value:** `AlwaysPullImages`

**Result:** Pass

### 1.1.12 - Ensure that the admission control plugin `DenyEscalatingExec` is set (Scored)

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--enable-admission-plugins=.*(DenyEscalatingExec).*").captures[].string'
```

**Returned Value:** `DenyEscalatingExec`

**Result:** Pass

### 1.1.13 - Ensure that the admission control plugin `SecurityContextDeny` is set (Scored)

#### Notes

This control may be out of date. This **SHOULD NOT** be set if you are using `PodSecurityPolicy` (PSP). From the Kubernetes 1.11 documentation:

> This should be enabled if a cluster doesn’t utilize pod security policies to restrict the set of values a security context can take.

Several system services (such as `nginx-ingress`) utilize `SecurityContext` to switch users and assign capabilities. These exceptions to the general principle of not allowing privilege or capabilities can be managed with PSP.

#### Audit

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--enable-admission-plugins=.*(SecurityContextDeny).*").captures[].string'
```

**Returned Value:** `null`

**Result:** Document

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

#### Audit

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--enable-admission-plugins=.*(NamespaceLifecycle).*").captures[].string'
```

**Returned Value:** `NamespaceLifecycle`

**Result:** Pass

### 1.1.15 - Ensure that the `--audit-log-path` argument is set as appropriate (Scored)

#### Notes

This path is the path inside of the container. It's combined with the RKE `cluster.yml` `extra-binds:` option to map the audit log to the host filesystem.

Audit logs should be collected and shipped off-system to guarantee their integrity.

#### Audit

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--audit-log-path=/var/log/kube-audit/audit-log.json").string'
```

**Returned Value:** `--audit-log-log=/var/log/kube-audit/audit-log.json`

**Result:** Pass

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

#### Notes

Audit logs should be collected and shipped off-system to guarantee their integrity. Rancher Labs recommends setting this argument to a low value to prevent audit logs from filling the local disk.

#### Audit

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--audit-log-maxage=\\d+").string'
```

**Returned Value:** `--audit-log-maxage=5`

**Result:** Pass

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

#### Notes

Audit logs should be collected and shipped off-system to guarantee their integrity. Rancher Labs recommends setting this argument to a low value to prevent audit logs from filling the local disk.

#### Audit

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--audit-log-maxbackup=\\d+").string'
```

**Returned Value:** `--audit-log-maxbackup=5`

**Result:** Pass

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

**Notes**

Audit logs should be collected and shipped off-system to guarantee their integrity.

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--audit-log-maxsize=\\d+").string'
```

**Returned Value:** `--audit-log-maxsize=100`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--authorization-mode=(Node|RBAC|,)+").string'
```

**Returned Value:** `--authorization-mode=Node,RBAC`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--token-auth-file=.*").string'
```

**Returned Value:** `null`

**Result:** Pass

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

**Notes**

RKE is using the kubelet's ability to automatically create self-signed certs. No CA cert is saved to verify the communication between `kube-apiserver` and `kubelet`.

**Mitigation**

Make sure nodes with `role:controlplane` are on the same local network as your nodes with `role:worker`. Use network ACLs to restrict connections to the kubelet port (10250/tcp) on worker nodes, only permitting it from controlplane nodes.

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--kubelet-certificate-authority=.*").string'
```

**Returned Value:** none

**Result:** Fail (See Mitigation)

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

**Audit (`--kubelet-client-certificate`)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--kubelet-client-certificate=.*").string'
```

**Returned Value:** `--kubelet-client-certificate=/etc/kubernetes/ssl/kube-apiserver.pem`

**Audit (`--kubelet-client-key`)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--kubelet-client-key=.*").string'
```

**Returned Value:** `--kubelet-client-key=/etc/kubernetes/ssl/kube-apiserver-key.pem`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--service-account-lookup=true").string'
```

**Returned Value:** `--service-account-lookup=true`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--enable-admission-plugins=.*(PodSecurityPolicy).*").captures[].string'
```

**Returned Value:** `PodSecurityPolicy`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--service-account-key-file=.*").string'
```

**Returned Value:** `--service-account-key-file=/etc/kubernetes/ssl/kube-apiserver-key.pem`

**Result:** Pass

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

**Audit (`--etcd-certfile`)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--etcd-certfile=.*").string'
```

**Returned Value:** `--etcd-certfile=/etc/kubernetes/ssl/kube-node.pem`

**Audit (`--etcd-keyfile`)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--etcd-keyfile=.*").string'
```

**Returned Value:** `--etcd-keyfile=/etc/kubernetes/ssl/kube-node-key.pem`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--enable-admission-plugins=.*(ServiceAccount).*").captures[].string'
```

**Returned Value:** `ServiceAccount`

**Result:** Pass

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

**Audit (`--tls-cert-file`)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cert-file=.*").string'
```

**Returned Value:** `--tls-cert-file=/etc/kubernetes/ssl/kube-apiserver.pem`

**Audit (`--tls-key-file`)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-private-key-file=.*").string'
```

**Returned Value:** `--tls-private-key-file=/etc/kubernetes/ssl/kube-apiserver-key.pem`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--client-ca-file=.*").string'
```

**Returned Value:** `--client-ca-file=/etc/kubernetes/ssl/kube-ca.pem` **Result:** Pass

### 1.1.30 - Ensure that the API Server only makes use of strong cryptographic ciphers (Not Scored)

**Audit (Allowed Ciphers)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256).*").captures[].string'
```

**Returned Value:**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384).*").captures[].string'
```

**Returned Value:**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305).*").captures[].string'
```

**Returned Value:**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_RSA_WITH_AES_128_GCM_SHA256).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_RSA_WITH_AES_256_GCM_SHA384).*").captures[].string'
```

**Returned Value:** `TLS_RSA_WITH_AES_256_GCM_SHA384`

**Audit (Disallowed Ciphers)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(CBC).*").captures[].string'
```

**Returned Value:** `null`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(RC4).*").captures[].string'
```

**Returned Value:** `null`

**Result:** Pass

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

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--etcd-cafile=.*").string'
```

**Returned Value:** `--etcd-cafile=/etc/kubernetes/ssl/kube-ca.pem`

**Result:** Pass

### 1.1.32 - Ensure that the `--authorization-mode` argument is set to Node (Scored)

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--authorization-mode=(Node|RBAC|,)+").string'
```

**Returned Value:** `--authorization-mode=Node,RBA` **Result:** Pass

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

#### Audit

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--enable-
admission-plugins=.*(NodeRestriction).*").captures[].string'
```

**Returned Value:** `NodeRestriction`

**Result:** Pass

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

#### Audit

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--
experimental-encryption-provider-config=.*").string'
```

**Returned Value:** `--experimental-encryption-provider-config=/etc/kubernetes/encryption.yaml`

**Result:** Pass

### 1.1.35 - Ensure that the encryption provider is set to aescbc (Scored)

#### Notes

Only the first provider in the list is active.

#### Audit

```
grep -A 1 providers: /etc/kubernetes/encryption.yaml | grep aescbc
```

**Returned Value:** `- aescbc:`

**Result:** Pass

### 1.1.36 - Ensure that the admission control plugin `EventRateLimit` is set (Scored)

#### Notes

The `EventRateLimit` plugin requires setting the `--admission-control-config-file` option and configuring details in the following files:

- `/etc/kubernetes/admission.yaml`
- `/etc/kubernetes/event.yaml`

See Host Configuration for details.

**Audit (Admissions plugin)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--enable-admission-plugins=.*(EventRateLimit).*").captures[].string'
```

**Returned Value:** `EventRateLimit`

**Audit (`--admission-control-config-file`)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--admission-control-config-file=.*").string'
```

**Returned Value:** `--admission-control-config-file=/etc/kubernetes/admission.yaml`

**Result:** Pass

### 1.1.37 Ensure that the AdvancedAuditing argument is not set to false (Scored)

**Notes**

`AdvancedAuditing=false` should not be set, but `--audit-policy-file` should be set and configured. See Host Configuration for a sample audit policy file.

**Audit (Feature Gate)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--feature-gates=.*(AdvancedAuditing=false).*").captures[].string'
```

**Returned Value:** `null`

**Audit (Audit Policy File)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--audit-policy-file=.*").string'
```

**Returned Value:** `--audit-policy-file=/etc/kubernetes/audit.yaml`

**Result:** Pass

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

**Notes**

RKE uses the default value of 60s and doesn't set this option.

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--request-timeout=.*").string'
```

**Returned Value:** `null`

**Result:** Pass

### 1.1.39 - Ensure that the API Server only makes use of strong cryptographic ciphers (Not Scored)

**Notes**

This appears to be a repeat of 1.1.30.

**Audit (Allowed Ciphers)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256).*").captures[].string'
```

**Returned Value:**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384).*").captures[].string'
```

**Returned Value:**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305).*").captures[].string'
```

**Returned Value:**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_RSA_WITH_AES_128_GCM_SHA256).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_RSA_WITH_AES_256_GCM_SHA384).*").captures[].string'
```

**Returned Value:** `TLS_RSA_WITH_AES_256_GCM_SHA384`

**Audit (Disallowed Ciphers)**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(CBC).*").captures[].string'
```

**Returned Value:** `null`

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(RC4).*").captures[].string'
```

**Returned Value:** `null`

**Result:** Pass

## 1.2 - Scheduler

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

**Audit**

```
docker inspect kube-scheduler | jq -e '.[0].Args[] | match("--profiling=false").string'
```

**Returned Value:** `--profiling=false` **Result:** Pass

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

**Audit**

```
docker inspect kube-scheduler | jq -e '.[0].Args[] | match("--address=127\\.0\\.0\\.1").string'
```

**Returned Value:** `--address=127.0.0.1` **Result:** Pass

### 1.3 - Controller Manager

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

**Audit**

```
docker inspect kube-controller-manager | jq -e '.[0].Args[] | match("--terminated-pod-gc-threshold=\\d+").string'
```

**Returned Value:** `--terminated-pod-gc-threshold=1000` **Result:** Pass

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

**Audit**

```
docker inspect kube-controller-manager | jq -e '.[0].Args[] | match("--profiling=false").string'
```

**Returned Value:** `--profiling=false`

**Result:** Pass

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

**Audit**

```
docker inspect kube-controller-manager | jq -e '.[0].Args[] | match("--
use-service-account-credentials=true").string'
```

**Returned Value:** `--use-service-account-credentials=true`

**Result:** Pass

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

**Audit**

```
docker inspect kube-controller-manager | jq -e '.[0].Args[] | match("--
service-account-private-key-file=.*").string'
```

**Returned Value:** `--service-account-private-key-file=/etc/kubernetes/ssl/kube-apiserver-key.pem`

**Result:** Pass

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

**Audit**

```
docker inspect kube-controller-manager | jq -e '.[0].Args[] | match("--
root-ca-file=.*").string'
```

**Returned Value:** `--root-ca-file=/etc/kubernetes/ssl/kube-ca.pem`

**Result:** Pass

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

**Notes**

RKE does not yet support certificate rotation. This feature is due for the 0.1.12 release of RKE.

**Audit**

```
docker inspect kube-apiserver | jq -e '.[0].Args[] | match("--feature-
gates=.*(RotateKubeletServerCertificate=true).*").captures[].string'
```

**Returned Value:** `null`

**Result:** Fail

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

**Audit**

```
docker inspect kube-controller-manager | jq -e '.[0].Args[] | match("--
address=127\\.0\\.0\\.1").string'
```

**Returned Value:** `--address=127.0.0.1`

**Result:** Pass

### 1.4 - Configuration Files

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

**Notes**

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

**Result:** Pass (Not Applicable)

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

**Notes**

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

**Result:** Pass (Not Applicable)

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

**Notes**

RKE doesn't require or maintain a configuration file for `kube-controller-manager`. All configuration is passed in as arguments at container run time.

**Result:** Pass (Not Applicable)

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

**Notes**

RKE doesn't require or maintain a configuration file for `kube-controller-manager`. All configuration is passed in as arguments at container run time.

**Result:** Pass (Not Applicable)

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

**Notes**

RKE doesn't require or maintain a configuration file for `kube-scheduler`. All configuration is passed in as arguments at container run time.

**Result:** Pass (Not Applicable)

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

**Notes**

RKE doesn't require or maintain a configuration file for kube-scheduler. All configuration is passed in as arguments at container run time.

**Result:** Pass (Not Applicable)

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

**Notes**

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

**Result:** Pass (Not Applicable)

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

**Notes**

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

**Result:** Pass (Not Applicable)

### 1.4.9 - Ensure that the Container Network Interface file permissions are set to `644` or more restrictive (Not Scored)

**Notes**

This is a manual check.

**Audit (`/var/lib/cni/networks/k8s-pod-network`)**

```
ls -l /var/lib/cni/networks/k8s-pod-network/
```

**Returned Value:**

```
-rw-r--r-- 1 root root   64 Nov  6 20:05 10.42.0.2
-rw-r--r-- 1 root root   64 Nov  6 20:05 10.42.0.3
-rw-r--r-- 1 root root   64 Nov  6 20:05 10.42.0.4
-rw-r--r-- 1 root root   64 Nov  6 20:05 10.42.0.5
-rw-r--r-- 1 root root   10 Nov  7 16:25 last_reserved_ip.0
```

**Audit (`/etc/cni/net.d`)**

```
ls -l /etc/cni/net.d/
```

**Returned Value:**

```
-rw-r--r-- 1 root root 1474 Nov  6 20:05 10-calico.conflist
-rw------- 1 root root 2529 Nov  6 20:05 calico-kubeconfig
```

**Result:** Pass

### 1.4.10 - Ensure that the Container Network Interface file ownership is set to `root:root` (Not Scored)

**Notes**

This is a manual check.

**Audit (`/var/lib/cni/networks/k8s-pod-network`)**

```
ls -l /var/lib/cni/networks/k8s-pod-network/
```

**Returned Value:**

```
-rw-r--r-- 1 root root   64 Nov  6 20:05 10.42.0.2
-rw-r--r-- 1 root root   64 Nov  6 20:05 10.42.0.3
-rw-r--r-- 1 root root   64 Nov  6 20:05 10.42.0.4
-rw-r--r-- 1 root root   64 Nov  6 20:05 10.42.0.5
-rw-r--r-- 1 root root   10 Nov  7 16:25 last_reserved_ip.0
```

**Audit (`/etc/cni/net.d`)**

```
ls -l /etc/cni/net.d/
```

**Returned Value:**

```
-rw-r--r-- 1 root root 1474 Nov  6 20:05 10-calico.conflist
-rw------- 1 root root 2529 Nov  6 20:05 calico-kubeconfig
```

**Result:** Pass

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

**Notes**

Files underneath the data dir are permissioned `700`

```
ls -al /var/lib/etcd
total 12
drwxr-xr-x  3 root root 4096 Nov 14 17:06 .
drwxr-xr-x 47 root root 4096 Nov 14 17:06 ..
drwx------  4 root root 4096 Nov 14 17:06 member
```

**Audit**

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

**Returned Value:** `755`

**Result:** Fail

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

**Notes**

The `etcd` container runs as the `root` user. The data directory and files are owned by `root`.

**Audit**

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

**Returned Value:** `root:root`

**Result:** Fail

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

**Notes**

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.

**Result:** Pass (Not Applicable)

### 1.4.14 - Ensure that ownership of `admin.conf` is set to `root:root` (Scored)

**Notes**

RKE does not store the default `kubectl` config credentials file on the nodes. It presents credentials to the user when `rke` is first run, and only on the device where the user ran the command. Rancher Labs recommends that this `kube_config_cluster.yml` file be kept in secure store.

**Result:** Pass (Not Applicable)

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

**Audit**

```
stat -c %a /etc/kubernetes/ssl/kubecfg-kube-scheduler.yaml
```

**Returned Value:** `644`

**Result:** Pass

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

**Audit**

```
stat -c %U:%G /etc/kubernetes/ssl/kubecfg-kube-scheduler.yaml
```

**Returned Value:** `root:root`

**Result:** Pass

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

**Audit**

```
stat -c %a /etc/kubernetes/ssl/kubecfg-kube-controller-manager.yaml
```

**Returned Value:** `644`

**Result:** Pass

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

**Audit**

```
stat -c %U:%G /etc/kubernetes/ssl/kubecfg-kube-controller-manager.yaml
```

**Returned Value:** `root:root`

**Result:** Pass

## 1.5 - etcd

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

**Audit `(--cert-file`)**

```
docker inspect etcd | jq -e '.[0].Args[] | match("--cert-file=.*").string'
```

**Returned Value:** `--cert-file=/etc/kubernetes/ssl/kube-etcd-172-31-22-135.pem`

**Audit (`--key-file`)**

```
docker inspect etcd | jq -e '.[0].Args[] | match("--key-file=.*").string'
```

**Returned Value:** `--key-file=/etc/kubernetes/ssl/kube-etcd-172-31-22-135-key.pem`

**Result:** Pass

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

**Notes**

Setting "--client-cert-auth" is the equivalent of setting "--client-cert-auth=true".

**Audit**

```
docker inspect etcd | jq -e '.[0].Args[] | match("--client-cert-auth(=true)*").string'
```

**Returned Value:** `--client-cert-auth`

**Result:** Pass

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

**Audit**

```
docker inspect etcd | jq -e '.[0].Args[] | match("--auto-tls(?:(?!=false).*)").string'
```

**Returned Value:** `null`

**Result:** Pass

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

**Audit (`--peer-cert-file)**

```
docker inspect etcd | jq -e '.[0].Args[] | match("--peer-cert-file=.*").string'
```

**Returned Value:** `--peer-cert-file=/etc/kubernetes/ssl/kube-etcd-172-31-22-135.pem`

**Audit (`--peer-key-file)**

```
docker inspect etcd | jq -e '.[0].Args[] | match("--peer-key-file=.*").string'
```

**Returned Value:** `--peer-key-file=/etc/kubernetes/ssl/kube-etcd-172-31-22-135-key.pem`

**Result:** Pass

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

**Notes**

Setting "--peer-client-cert-auth" is the equivalent of setting "--peer-client-cert-auth=true".

**Audit**

```
docker inspect etcd | jq -e '.[0].Args[] | match("--peer-client-cert-auth(=true)*").string'
```

**Returned Value:** `--client-cert-auth`

**Result:** Pass

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

**Audit**

```
docker inspect etcd | jq -e '.[0].Args[] | match("--peer-auto-tls(?:(?!=false).*)").string'
```

**Returned Value:** `null`

**Result:** Pass

### 1.5.7 - Ensure that a unique Certificate Authority is used for `etcd` (Not Scored)

**Notes**

RKE does not currently implement a seperate CA for etcd certificates.

`--trusted-ca-file` is set and different from the `--client-ca-file` used by `kube-apiserver`.

**Audit**

```
docker inspect etcd | jq -e '.[0].Args[] | match("--trusted-ca-file=(?:(?!/etc/kubernetes/ssl/kube-ca.pem).*)").string'
```

**Returned Value:** `null`

**Result:** Fail

## 1.6 - General Security Primitives

These "Not Scored" controls are implementation best practices. To ease the administrative burden, we recommend that you implement these best practices on your workload clusters by creating clusters with Rancher rather than using RKE alone.

### 1.6.1 - Ensure that the cluster-admin role is only used where required (Not Scored)

> Rancher has built in support for maintaining and enforcing Kubernetes RBAC on your workload clusters.

Rancher has the ability integrate with external authentication sources (LDAP, SAML, AD…) allows easy access with unique credentials to your existing users or groups.

### 1.6.2 - Create administrative boundaries between resources using namespaces (Not Scored)

With Rancher, users or groups can be assigned access to all clusters, a single cluster or a "Project" (a group of one or more namespaces in a cluster). This allows granular access control to cluster resources.

### 1.6.3 - Create network segmentation using Network Policies (Not Scored)

Rancher can (optionally) automatically create Network Policies to isolate "Projects" (a group of one or more namespaces) in a cluster.

See "Cluster Options" when creating a cluster with Rancher to turn on Network Isolation.

### 1.6.4 - Ensure that the `seccomp` profile is set to `docker/default` in your pod definitions (Not Scored)

Since this requires the enabling of AllAlpha feature gates we would not recommend enabling this feature at the moment.

### 1.6.5 - Apply security context to your pods and containers (Not Scored)

This practice does go against control 1.1.13, but we prefer using PSP and allowing security context to be set over a blanket deny.

Rancher allows users to set various Security Context options when launching pods via the GUI interface.

### 1.6.6 - Configure image provenance using the `ImagePolicyWebhook` admission controller (Not Scored)

Image Policy Webhook requires a 3rd party service to enforce policy. This can be configured in the `--admission-control-config-file`. See the Host configuration section for the admission.yaml file.

### 1.6.7 - Configure network policies as appropriate (Not Scored)

Rancher can (optionally) automatically create Network Policies to isolate projects (a group of one or more namespaces) within a cluster.

See the ***Cluster Options*** section when creating a cluster with Rancher to turn on network isolation.

### 1.6.8 - Place compensating controls in the form of PSP and RBAC for privileged container usage (Not Scored)

Section 1.7 of this guide shows how to add and configure a default "restricted" PSP based on controls.

With Rancher you can create a centrally maintained "restricted" PSP and deploy it to all of the clusters that Rancher manages.

### 1.7 - Pod Security Policies (PSP)

This RKE configuration has two Pod Security Policies.

- `default-psp`: assigned to namespaces that require additional privileged access: `kube-system`, `ingress-nginx` and `cattle-system`.
- `restricted`: This is the cluster default PSP and follows the best practices defined by controls in this section.

### 1.7.1 - Do not admit privileged containers (Not Scored)

**Notes**

The restricted PSP is available to all ServiceAccounts.

**Audit**

```
kubectl get psp restricted -o jsonpath='{.spec.privileged}' | grep "true"
```

**Returned Value:** `null`

**Result:** Pass

### 1.7.2 - Do not admit containers wishing to share the host process ID namespace (Scored)

**Notes**

The restricted PSP is available to all ServiceAccounts.

**Audit**

```
kubectl get psp restricted -o jsonpath='{.spec.hostPID}' | grep "true"
```

**Returned Value:** `null`

**Result:** Pass

### 1.7.3 - Do not admit containers wishing to share the host IPC namespace (Scored)

**Notes**

The restricted PSP is available to all ServiceAccounts.

**Audit**

```
kubectl get psp restricted -o jsonpath='{.spec.hostIPC}' | grep "true"
```

**Returned Value:** `null`

**Result:** Pass

### 1.7.4 - Do not admit containers wishing to share the host network namespace (Scored)

**Notes**

The restricted PSP is available to all ServiceAccounts.

**Audit**

```
kubectl get psp restricted -o jsonpath='{.spec.hostNetwork}' | grep "true"
```

**Returned Value:** `null`

**Result:** Pass

### 1.7.5 - Do not admit containers with `allowPrivilegeEscalation` (Scored)

**Notes**

The restricted PSP is available to all ServiceAccounts.

**Audit**

```
kubectl get psp restricted -o jsonpath='{.spec.allowPrivilegeEscalation}'
| grep "true"
```

**Returned Value:** `null`

**Result:** Pass

### 1.7.6 - Do not admit containers whose processes run as `root` (Not Scored)

**Notes**

The restricted PSP is available to all ServiceAccounts.

**Audit**

```
kubectl get psp restricted -o jsonpath='{.spec.runAsUser.rule}' | grep
"RunAsAny"
```

**Returned Value:** `null`

**Result:** Pass

### 1.7.7 - Do not admit containers with dangerous capabilities (Not Scored)

**Notes**

The restricted PSP is available to all ServiceAccounts.

**Audit**

```
kubectl get psp restricted -o jsonpath='{.spec.requiredDropCapabilities}'
| grep "NET_RAW"
```

**Returned Value:** `[NET_RAW]`

**Result:** Pass

# 2 - Worker Node Security Configuration

## 2.1 - Kubelet

### 2.1.1 - Ensure that the `--allow-privileged` argument is set to false (Scored)

**Notes**

The `--allow-privileged` argument is deprecated from Kubernetes v1.11, and the default setting is `true` with the intention that users should use `PodSecurityPolicy` settings to allow or prevent privileged containers.

Our RKE configuration uses `PodSecurityPolicy` with a default policy to reject privileged containers.

**Result:** Pass (Not Applicable)

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

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--anonymous-
auth=false").string'
```

**Returned Value:** `--anonymous-auth=false`

**Result:** Pass

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

**Notes**

RKE currently runs the kubelet without the `--authorization-mode` flag.

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--authorization-mode=Webhook").string'
```

**Returned Value:** `null`

**Result:** Fail

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

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--client-ca-file=.*").string'
```

**Returned Value:** `--client-ca-file=/etc/kubernetes/ssl/kube-ca.pem`

**Result:** Pass

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

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--read-only-port=0").string'
```

**Returned Value:** `--read-only-port=0`

**Result:** Pass

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

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--streaming-connection-idle-timeout=.*").string'
```

**Returned Value:** `--streaming-connection-idle-timeout=1800s`

**Result:** Pass

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

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--protect-kernel-defaults=true").string'
```

**Returned Value:** `--protect-kernel-defaults=true`

**Result:** Pass

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

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--make-iptables-util-chains=true").string'
```

**Returned Value:** `--make-iptables-util-chains=true`

**Result:** Pass

### 2.1.9 - Ensure that the `--hostname-override` argument is not set (Scored)

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--hostname-override=.*").string'
```

**Returned Value:** `--hostname-override=<ipv4 address>`

**Result:** Fail

### 2.1.10 - Ensure that the `--event-qps` argument is set to `0` (Scored)

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--event-qps=0").string'
```

**Returned Value:** `--event-qps=0`

**Result:** Pass

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

**Notes**

RKE does not set these options and uses the kubelet's self generated certificates for TLS communication. These files are located in the default directory (`/var/lib/kubelet/pki`).

**Audit (`--tls-cert-file`)**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cert-
file=.*").string'
```

**Returned Value:** `null`

**Audit (`--tls-private-key-file`)**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-private-key-
file=.*").string'
```

**Returned Value:** `null`

**Result:** Pass

### 2.1.12 - Ensure that the `--cadvisor-port` argument is set to `0` (Scored)

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--cadvisor-
port=0").string'
```

**Returned Value:** `--cadvisor-port=0`

**Result:** Pass

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

**Notes**

RKE will enable certificate rotation in version 0.1.12.

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--rotate-certificates=true").string'
```

**Returned Value:** `null`

**Result:** Fail

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

#### Notes

RKE does not yet support certificate rotation. This feature is due for the 0.1.12 release of RKE.

#### Audit

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--feature-gates=.*(RotateKubeletServerCertificate=true).*").captures[].string'
```

**Returned Value:** `null`

**Result:** Fail

### 2.1.15 - Ensure that the kubelet only makes use of strong cryptographic ciphers (Not Scored)

#### Audit (Allowed Ciphers)

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`

#### Audit

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`

#### Audit

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-
suites=.*(TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305`

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-
suites=.*(TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256).*").captures[].string'
```

**Returned Value:**

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-
suites=.*(TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384).*").captures[].string'
```

**Returned Value:**

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-
suites=.*(TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305).*").captures[].string'
```

**Returned Value:**

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-
suites=.*(TLS_RSA_WITH_AES_128_GCM_SHA256).*").captures[].string'
```

**Returned Value:** `TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305`

**Audit**

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-
suites=.*(TLS_RSA_WITH_AES_256_GCM_SHA384).*").captures[].string'
```

**Returned Value:** `TLS_RSA_WITH_AES_256_GCM_SHA384`

#### Audit (Disallowed Ciphers)

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(CBC).*").captures[].string'
```

**Returned Value:** `null`

#### Audit

```
docker inspect kubelet | jq -e '.[0].Args[] | match("--tls-cipher-suites=.*(RC4).*").captures[].string'
```

**Returned Value:** `null`

**Result:** Pass

## 2.2 - Configuration Files

### 2.2.1 - Ensure that the permissions for `kubelet.conf` are set to `644` or more restrictive (Scored)

#### Notes

This is the value of the `--kubeconfig` option.

#### Audit

```
stat -c %a /etc/kubernetes/ssl/kubecfg-kube-node.yaml
```

**Returned Value:** `644`

**Result:** Pass

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

#### Notes

This is the value of the `--kubeconfig` option.

#### Audit

```
stat -c %U:%G /etc/kubernetes/ssl/kubecfg-kube-node.yaml
```

**Returned Value:** `root:root`

**Result:** Pass

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

#### Notes

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

**Result:** Pass (Not Applicable)

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

#### Notes

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

**Result:** Pass (Not Applicable)

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

#### Audit

```
stat -c %a /etc/kubernetes/ssl/kubecfg-kube-proxy.yaml
```

**Returned Value:** `644`

**Result:** Pass

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

#### Audit

```
stat -c %U:%G /etc/kubernetes/ssl/kubecfg-kube-proxy.yaml
```

**Returned Value:** `root:root`

**Result:** Pass

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

#### Audit

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

**Returned Value:** `644`

**Result:** Pass

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

**Audit**

```
stat -c %U:%G /etc/kubernetes/ssl/kube-ca.pem
```

**Returned Value:** `root:root`

**Result:** Pass

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

**Notes**

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

**Result:** Pass (Not Applicable)

### 2.2.10 - Ensure that the kubelet configuration file permissions are set to `644` or more restrictive (Scored)

**Notes**

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

**Result:** Pass (Not Applicable)