www.atmicnetworks.com

ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 1

SIMULATION

Create a new ServiceAccount named backend-sa in the existing namespace default, which has the capability to list the pods inside the namespace default.

Create a new Pod named backend-pod in the namespace default, mount the newly created sa backend-sa to the pod, and Verify that the pod is able to list pods.

Ensure that the Pod is running.

Answer: See the Explanation below:

Explanation:

A service account provides an identity for processes that run in a Pod.

When you (a human) access the cluster (for example, using kubectl), you are authenticated by the apiserver as a particular User Account (currently this is usually admin, unless your cluster administrator has customized your cluster). Processes in containers inside pods can also contact the apiserver. When they do, they are authenticated as a particular Service Account (for example, default).

When you create a pod, if you do not specify a service account, it is automatically assigned the default service account in the same namespace. If you get the raw json or yaml for a pod you

have created (for example, kubectl get pods/<podname> -o yaml), you can see the spec.serviceAccountName field has been automatically set.

You can access the API from inside a pod using automatically mounted service account credentials, as described in Accessing the Cluster. The API permissions of the service account depend on the authorization plugin and policy in use.

In version 1.6+, you can opt out of automounting API credentials for a service account by setting automountServiceAccountToken: false on the service account:

apiVersion: v1

kind: ServiceAccount

www.atmicnetworks.com

metadata:

name: build-robot

automountServiceAccountToken: false

...

In version 1.6+, you can also opt out of automounting API credentials for a particular pod:

apiVersion: v1

kind: Pod

metadata:

name: my-pod

spec:

serviceAccountName: build-robot

automountServiceAccountToken: false

The pod spec takes precedence over the service account if both specify a automountServiceAccountToken value.

Question: 2

SIMULATION

Fix all issues via configuration and restart the affected components to ensure the new setting takes effect.

Fix all of the following violations that were found against the API server:-

a. Ensure the --authorization-mode argument includes RBAC

b. Ensure the --authorization-mode argument includes Node

c. Ensure that the --profiling argument is set to false

Fix all of the following violations that were found against the Kubelet:-

a. Ensure the --anonymous-auth argument is set to false.

b. Ensure that the --authorization-mode argument is set to Webhook.

Fix all of the following violations that were found against the ETCD:-

a. Ensure that the --auto-tls argument is not set to true

Hint: Take the use of Tool Kube-Bench

Answer: See the Explanation below.

Explanation:

API server:

Ensure the --authorization-mode argument includes RBAC

Turn on Role Based Access Control.
Role Based Access Control (RBAC) allows fine-grained control over the operations that different entities can perform on different objects in the cluster. It is recommended to use the RBAC authorization mode.

Fix - Buildtime

Kubernetes

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

component: kube-apiserver

tier: control-plane

name: kube-apiserver

namespace: kube-system

spec:

containers:

- command:

www.atmicnetworks.com

```
+ - kube-apiserver
+ - --authorization-mode=RBAC,Node
    image: gcr.io/google_containers/kube-apiserver-amd64:v1.6.0
    livenessProbe:
     failureThreshold: 8
     httpGet:
      host: 127.0.0.1
      path: /healthz
      port: 6443
      scheme: HTTPS
     initialDelaySeconds: 15
     timeoutSeconds: 15
    name: kube-apiserver-should-pass
    resources:
     requests:
      cpu: 250m
    volumeMounts:
    - mountPath: /etc/kubernetes/ name: k8s
     readOnly: true
    - mountPath: /etc/ssl/certs name: certs
    -   mountPath: /etc/pki
     name: pki
hostNetwork: true
volumes:
-   hostPath:
     path: /etc/kubernetes
```

www.atmicnetworks.com

name: k8s

- hostPath:

path: /etc/ssl/certs

name: certs

- hostPath:

path: /etc/pki

name: pki

Ensure the --authorization-mode argument includes Node

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'

Ensure that the --profiling argument is set to false

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'

www.atmicnetworks.com

Fix all of the following violations that were found against the Kubelet:-

Ensure the --anonymous-auth argument is set to false.

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'

2) Ensure that the --authorization-mode argument is set to Webhook.

Audit

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

Returned Value: --authorization-mode=Webhook

Fix all of the following violations that were found against the ETCD:-

a. Ensure that the --auto-tls argument is not set to true

Do not use self-signed certificates for TLS. etcd is a highly-available key value store used by Kubernetes deployments for persistent storage of all of its REST API objects. These objects are sensitive in nature and should not be available to unauthenticated clients. You should enable the client authentication via valid certificates to secure the access to the etcd service.

Fix - Buildtime

www.atmicnetworks.com

Kubernetes

apiVersion: v1

kind: Pod

metadata:

annotations:

scheduler.alpha.kubernetes.io/critical-pod: ""

creationTimestamp: null

labels:

component: etcd

tier: control-plane

name: etcd

namespace: kube-system

spec:

containers:

- command:

+ - etcd

+ - --auto-tls=true

image: k8s.gcr.io/etcd-amd64:3.2.18

imagePullPolicy: IfNotPresent

livenessProbe:

exec:

command:

- /bin/sh

- -ec

- ETCDCTL_API=3 etcdctl --endpoints=https://[192.168.22.9]:2379 --

cacert=/etc/kubernetes/pki/etcd/ca.crt

- --cert=/etc/kubernetes/pki/etcd/healthcheck-client.crt --

www.atmicnetworks.com

```
key=/etc/kubernetes/pki/etcd/healthcheck-client.key
      get foo
    failureThreshold: 8
    initialDelaySeconds: 15
    timeoutSeconds: 15
  name: etcd-should-fail
  resources: {}
  volumeMounts:
  - mountPath: /var/lib/etcd
    name: etcd-data
  - mountPath: /etc/kubernetes/pki/etcd
    name: etcd-certs
hostNetwork: true
priorityClassName: system-cluster-critical volumes:
- hostPath:
    path: /var/lib/etcd
    type: DirectoryOrCreate
  name: etcd-data
- hostPath:
    path: /etc/kubernetes/pki/etcd
    type: DirectoryOrCreate
  name: etcd-certs
status: {}
```

Explanation:

www.atmicnetworks.com

```
candidate@cli:-$ kubectl delete sa/podrunner -n qa
serviceaccount "podrunner" deleted
candidate@cli:~$ kubectl config use-context KSCS00201
Switched to context "KSCS0O2O1".
candida te@d i: ~$ ssh kscs00201-master
Warning: Permanently added '10.240.86.194' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in
the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

root8kscs00201-master:~l vim /etc/kubernetes/manifests/tube-apiserver.yaml
rootskscs00201-master:-I systemctl daemon-reload
root@kscs00201-master:~* systemctl restart kubelet.service
root@kscs00201-master:'* systemctl enable kubelet.service
root@kscs00201-master:-t systemctl status kubelet.service^
• kubelet.service - kubelet: The Kubernetes Node Agent
     Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled) Drop-In:
/etc/systemd/system/kubelet.service.d
             •—10-kubeadm.conf
     Active: active (running) since Eri 2022-05-20 14:19:31 UTC; 29s ago Docs: https://kubernetes.io/docs/home/
   Main PID: 134205 (kubelet)
      Tasks: 16 (limit: 76200)
     Memory: 39.5M
     CGroup: /system.slice/kubelet.service
             *—134205 /usr/bin/kubelet —bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubj

May 20 14:19:35 kscs00201~master kubelet[134205]: May 20 14:19:35 kscs00201-master kubelet[134205]: May 20 14:19:35 kscs00201 roaster kubelet[134205]: May 20 14:19:35 kscs00201-master kubelet[134205]: May 20 14:19:36 kscs00201-master kubelet[134205]: May 20 14:19:37 kscs00201-master kubelet[134205]: May 20 14:19:37 kscs00201-master kubelet[134205]: May 20 14:19:37 kscs00201-master kubelet[134205]: May 20 14:19:38 kscs00201-master kubelet[134205]: May 20 14:19:40 kscs00201 master kubelet[134205]:
limes i-22/22 (END)

10520 14:19:35.420825 134205 reconciler.
10520 14:19:35.420863 134205 reconciler.
10520 14:19:35.420907 134205 reconciler.
10520 14:19:35.420928 134205 reconciler.
10520 14:19:36.572353 134205 request.go:
10520 14:19:37.112347 134205 prober mana
E0520 14:19:37.185076 134205 kubelet.go:
10520 14:19:37.645798 134205 kubelet.go:
10520 14:19:38.184062 134205 kubelet.go:
10520 14:19:40.036042 134205 probermana

de Agent
et.service; enabled; vendor preset: enabled) ce.d

5-20 14:19:31 UTC; 29s ago

trap-kubeconfig=/etc/kubernetes/bootstrap-kubelet,conf —kubeconflg=/etc/kubernetes/kubelerj

5]: 10520 14:19:35.420825 5): 134205 reconciler.go:221] "operationExecutor.VerifyControllerAt.tB
10520 14:19:35.420863 5J: 134205 reconci ler .go: 221 ] " iperationExecutor.Veri fyControllerAt J
10520 14:19:35.420907 5[: 134205 reconciler.go:221) "operationExecutor.VerifyControllerAtJ
10520 14:19:35.420928 5]: 134205 reconciler.go:157) "Reconciler: start to sync state"
10520 14:19:36.572353 5]: 134205 request.go:665] Waited for 1.049946364s due to client-sic
10520 14:19:37.112347 5]: 134205 prober_manager.go:255] "Failed to trigger a manual run" f
E0520 14:19:37.185076 5]: 134205 kubelet.go:1711] "Failed creating a mirror pod for" err-"
10520 14:19:37.645798 5]: 134205 kubelet.go:1693) "Trying to delete pod" pod="kube-system/
10520 14:19:38.184062 5]: 134205 kubelet.go:1698j "Deleted mirror pod because it is outdat
10520 14:19:40.036042 134205 prcber_manager.go:255] "Failed to trigger a manual run" p

lines 1-22/22 (END)
let.conf —kubeconfig^/etc/kubernetes/kubelet.conf —config=/var/lib/kubelet/config.yaml —|

0:221] "operationExecutor.VerifyControllerAttachedVoltune started for volume \"kube-proxy!" I o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"lib-modules\"l o:221] "operationExecutor.VerifyControllerAttachedVolume started for volume \"flannel-cfg\"| 6:157] "Reconciler: start to sync state"
65] Waited for 1.049946364s due to client-side throttling, not priority and fairness, reguej er.go:255] "Failed to trigger a manual run" probe="Readiness"
711] "Failed creating a mirror pod for" err="pods \"kube-apiserver-kscs00201-master\" alreal 693] "Trying to delete pod"
```

```
pod="kube-system/kube-apiserver~kscs00201-master" podUID=bb91ell 698] "Deleted mirror pod because it is outdated"
pod="kubc system/kube apiserver kscs00201 I er.go:255] "Failed to trigger a manual run" probe="Readiness"

root@kscs00201-master:~# vim /var/lib/kubelet/config.yaml I
```


```
root@kscs00201-master:~$ vim /var/lib/kubelet/config.yaml
root0kscsOO2Ol-master:~i vim /var/lib/kubelet/config.yaml root@kscs00201-master:M
vim /etc/kubernetes/manifests/etcd.yaml root@kscs00201-master:~l systemctl daemon-
reload
root@kscs00201-master:~f systemctl restart kubelet.service
root@kscs00201-master:~i systemctl status kubelet.serviced
```

Question: 3

SIMULATION

Create a PSP that will prevent the creation of privileged pods in the namespace.

Create a new PodSecurityPolicy named prevent-privileged-policy which prevents the creation of privileged pods.

Create a new ServiceAccount named psp-sa in the namespace default.

Create a new ClusterRole named prevent-role, which uses the newly created Pod Security Policy prevent-privileged-policy.

Create a new ClusterRoleBinding named prevent-role-binding, which binds the created ClusterRole prevent-role to the created SA psp-sa.

Also, Check the Configuration is working or not by trying to Create a Privileged pod, it should get failed.

Answer: See the Explanation below.

Explanation:

Create a PSP that will prevent the creation of privileged pods in the namespace.

www.atmicnetworks.com

```
$ cat clusterrole-use-privileged.yaml

apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: use-privileged-psp
rules:
- apiGroups: ['policy']
  resources: ['podsecuritypolicies']
  verbs:     ['use']
  resourceNames:
  - default-psp


apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
    name: privileged-role-bind
  namespace: psp-test
roleRef:
    apiGroup: rbac.authorization.k8s.io
    kind: ClusterRole
    name: use-privileged-psp
subjects:
- kind: ServiceAccount
    name: privileged-sa
$ kubectl -n psp-test apply -f clusterrole-use-privileged.yaml
```

www.atmicnetworks.com

After a few moments, the privileged Pod should be created.

Create a new PodSecurityPolicy named prevent-privileged-policy which prevents the creation of privileged pods.

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: example
spec:
  privileged: false # Don't allow privileged pods!
```

- The rest fills in some required fields.

```
  seLinux:
    rule: RunAsAny
  supplementalGroups:
    rule: RunAsAny
  runAsUser:
    rule: RunAsAny
  fsGroup:
    rule: RunAsAny
  volumes:
  -
```

And create it with kubectl:

```
kubectl-admin create -f example-psp.yaml
```

Now, as the unprivileged user, try to create a simple pod:

```
kubectl-user create -f- <<EOF
```

www.atmicnetworks.com

```
apiVersion: v1
kind: Pod
metadata:
name: pause
spec:
containers:
- name: pause
image: k8s.gcr.io/pause
EOF
```

The output is similar to this:

```
Error from server (Forbidden): error when creating "STDIN": pods "pause" is forbidden: unable to validate against any pod security policy: []
```

Create a new ServiceAccount named psp-sa in the namespace default.

```
$ cat clusterrole-use-privileged.yaml
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
name: use-privileged-psp
rules:
- apiGroups: ['policy']
resources: ['podsecuritypolicies']
verbs: ['use']
resourceNames:
```

www.atmicnetworks.com

```
- default-psp

---

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

  name: privileged-role-bind

namespace: psp-test

roleRef:

  apiGroup: rbac.authorization.k8s.io

  kind: ClusterRole

  name: use-privileged-psp

subjects:

- kind: ServiceAccount

  name: privileged-sa
```

$ kubectl -n psp-test apply -f clusterrole-use-privileged.yaml

After a few moments, the privileged Pod should be created.

Create a new ClusterRole named prevent-role, which uses the newly created Pod Security Policy prevent-privileged-policy.

```
apiVersion: policy/v1beta1

kind: PodSecurityPolicy

metadata:

name: example

spec:

privileged: false # Don't allow privileged pods!
```

www.atmicnetworks.com

- The rest fills in some required fields.

```
seLinux:
  rule: RunAsAny
supplementalGroups:
  rule: RunAsAny
runAsUser:
  rule: RunAsAny
fsGroup:
  rule: RunAsAny
volumes:
-
```

And create it with kubectl:

```
kubectl-admin create -f example-psp.yaml
```

Now, as the unprivileged user, try to create a simple pod:

```
kubectl-user create -f- <<EOF
apiVersion: v1
kind: Pod
metadata:
  name: pause
spec:
  containers:
    - name: pause
      image: k8s.gcr.io/pause
EOF
```

The output is similar to this:

```
Error from server (Forbidden): error when creating "STDIN": pods "pause" is forbidden: unable to validate
```

www.atmicnetworks.com

against any pod security policy: []

Create a new ClusterRoleBinding named prevent-role-binding, which binds the created ClusterRole prevent-role to the created SA psp-sa.

apiVersion: rbac.authorization.k8s.io/v1

This role binding allows "jane" to read pods in the "default" namespace.

You need to already have a Role named "pod-reader" in that namespace.

kind: RoleBinding

metadata:

name: read-pods

namespace: default

subjects:

You can specify more than one "subject"

kind: User

name: jane # "name" is case sensitive

apiGroup: rbac.authorization.k8s.io

roleRef:

"roleRef" specifies the binding to a Role / ClusterRole

kind: Role #this must be Role or ClusterRole

name: pod-reader # this must match the name of the Role or ClusterRole you wish to bind to

apiGroup: rbac.authorization.k8s.io

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

namespace: default

name: pod-reader

www.atmicnetworks.com

rules:

apiGroups: [""] # "" indicates the core API group

resources: ["pods"]

verbs: ["get", "watch", "list"]

Question: 4

SIMULATION

www.atmicnetworks.com

You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSCHOO	ksch00201	ksch00201
201	-master	-workerl

You can switch the cluster/configuration context using the following command:

```
[candidate@cli] $ kubec tl config use-context KS CH00201
```

Context

A Role bound to a Pod's ServiceAccount grants overly permissive permissions. Complete the following tasks to reduce the set of permissions.

Task

Given an existing Pod named web-pod running in the namespace security.

Edit the existing Role bound to the Pod's ServiceAccount sa-dev-1 to only allow performing watch operations, only on resources of type services.

Create a new Role named role-2 in the namespace security, which only allows performing update

operations, only on resources of type namespaces.

Create a new RoleBinding named role-2-binding binding the newly created Role to the Pod's ServiceAccount.

Don't delete the existing RoleBinding.

Answer: See explanation below.

Explanation:

```
candidate@cli:~$ kubectl config use-context KSCH00201 Switched to context "KSCH00201". candidategcii:~$ kubectl get pods -n
security NAME          READY       STATUS       RESTARTS          AGE
web-pod      1/1       Running      0                 6h9m
candidate@cli:~$ kubectl get deployments.apps -n security No resources found in security namespace.
candidateScli:~$ kubectl describe rolebindings.rbac.authorization.kBs.io -n security Name:      dev-role
Labels:              <none>
Annota t ions: <none> Role:
    Kind: Role
    Name: dev-role
Subjects:
    Kind                    Name Namespace

    ServiceAccount sa-dev-1
candidateScli:~5 kubectl describe role dev-role n security
Name:               dev-role
Labels:             <none>
Annotations: <none>
PolicyRule:
    Resources Non-Resource URLs Resource Names Verbs

    *                [J                        []                        [*]
candidate@cli:~$ kubectl edit role/dev-role -n security |
```

b4c9ddd6-2729-43bd-8fbd-b2d227f4c4cd

iGroupsi

services

- watch

```
candidate9cli:~$ kubectl describe role dev-role -n security
Name:             dev-role
Labels:           <none>
Annotations: <none>
PolicyRule:
    Resources Non-Resource URLs Resource Names Verbs

  *           [1              [1              [*l
candidate^cli:-$ kubectl edit role/dev-role -n security role.rbac.authorization.kBs.io/dev-role edited candidateScli:-? kubectl
describe role dev-role -n security Name:.  dev-role
Labels:           <nohe>
Annotations: <none> PolicyRule: Resources Non-Resource URLs Resource Names Verbs services []      []          [watch]
candidateScli:~5 kubectl get pods -h security NAME          READY   STATUS  RESTARTS          AGE
web-pod    1/1          Running      0                6hl2m
candidate@cli:-$ kubectl get pods/web-pod n Security -o yaml ( grep serviceAccount serviceAccount: sa-dev-1 servi
:eAccountjfame: sa-dev-1 - serviceAccountTOkeni
candidate@cli:~S kubectl create role role-2 --verb-update —resource-namespaces -n security role.rbac.authorization.kBs.io/role-2
created candidate@cli:'$ kubectl create rolebinding role-2-binding —role —role      —role=
candidate@cli:~5 kubectl create rolebinding role-2-binding - role=role-2 --serviceaccount=se curity:sa-dev-1 -n security
rolebinding.rbac.authorization. kBs.io/role-2-binding created candidatc@cli:~$ Q
```

Question: 5

SIMULATION

Enable audit logs in the cluster, To Do so, enable the log backend, and ensure that

1. logs are stored at /var/log/kubernetes-logs.txt.
2. Log files are retained for 12 days.
3. at maximum, a number of 8 old audit logs files are retained.
4. set the maximum size before getting rotated to 200MB

Edit and extend the basic policy to log:

1. namespaces changes at RequestResponse
2. Log the request body of secrets changes in the namespace kube-system.
3. Log all other resources in core and extensions at the Request level.
4. Log "pods/portforward", "services/proxy" at Metadata level.

5. Omit the Stage RequestReceived

All other requests at the Metadata level

Answer: See the explanation below:

Explanation:

Kubernetes auditing provides a security-relevant chronological set of records about a cluster. Kube- apiserver performs auditing. Each request on each stage of its execution generates an event, which is then pre-processed according to a certain policy and written to a backend. The policy determines what's recorded and the backends persist the records.

You might want to configure the audit log as part of compliance with the CIS (Center for Internet Security) Kubernetes Benchmark controls.

The audit log can be enabled by default using the following configuration in cluster.yml:

```
services:
kube-api:
  audit_log:
    enabled: true
```

When the audit log is enabled, you should be able to see the default values at /etc/kubernetes/audit-policy.yaml

The log backend writes audit events to a file in JSONlines format. You can configure the log audit backend using the following kube-apiserver flags:

- -audit-log-path specifies the log file path that log backend uses to write audit events. Not specifying this flag disables log backend. - means standard out
- -audit-log-maxage defined the maximum number of days to retain old audit log files
- -audit-log-maxbackup defines the maximum number of audit log files to retain
- -audit-log-maxsize defines the maximum size in megabytes of the audit log file before it gets rotated

If your cluster's control plane runs the kube-apiserver as a Pod, remember to mount the hostPath to the location of the policy file and log file, so that audit records are persisted. For example:

www.atmicnetworks.com

--audit-policy-file=/etc/kubernetes/audit-policy.yaml \

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

Question: 6

SIMULATION

Analyze and edit the given Dockerfile

FROM ubuntu:latest

RUN apt-get update -y

RUN apt-install nginx -y

COPY entrypoint.sh /

ENTRYPOINT ["/entrypoint.sh"]

USER ROOT

Fixing two instructions present in the file being prominent security best practice issues

Analyze and edit the deployment manifest file

apiVersion: v1

kind: Pod

metadata:

name: security-context-demo-2

spec:

securityContext:

runAsUser: 1000

www.atmicnetworks.com

containers:

name: sec-ctx-demo-2

image: gcr.io/google-samples/node-hello:1.0

securityContext:

runAsUser: 0

privileged: True

allowPrivilegeEscalation: false

Fixing two fields present in the file being prominent security best practice issues

Don't add or remove configuration settings; only modify the existing configuration settings

Whenever you need an unprivileged user for any of the tasks, use user test-user with the user id 5487

Answer: See the explanation below:

Explanation:

FROM debian:latest

MAINTAINER k@bogotobogo.com

1 - RUN

RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -yq apt-utils

RUN DEBIAN_FRONTEND=noninteractive apt-get install -yq htop

RUN apt-get clean

2 - CMD

#CMD ["htop"]

#CMD ["ls", "-l"]

www.atmicnetworks.com

```
# 3 - WORKDIR and ENV

WORKDIR /root

ENV DZ version1
```

$ docker image build -t bogodevops/demo .

```
Sending build context to Docker daemon 3.072kB

Step 1/7 : FROM debian:latest

 ---> be2868bebaba

Step 2/7 : MAINTAINER k@bogotobogo.com

 ---> Using cache

 ---> e2eef476b3fd

Step 3/7 : RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -yq apt-utils

 ---> Using cache

 ---> 32fd044c1356

Step 4/7 : RUN DEBIAN_FRONTEND=noninteractive apt-get install -yq htop

 ---> Using cache

 ---> 0a5b514a209e

Step 5/7 : RUN apt-get clean

 ---> Using cache

 ---> 5d1578a47c17

Step 6/7 : WORKDIR /root

 ---> Using cache

 ---> 6b1c70e87675

Step 7/7 : ENV DZ version1

 ---> Using cache

 ---> cd195168c5c7

Successfully built cd195168c5c7

Successfully tagged bogodevops/demo:latest
```

www.atmicnetworks.com

Question: 7
SIMULATION

Create a RuntimeClass named gvisor-rc using the prepared runtime handler named runsc.

Create a Pods of image Nginx in the Namespace server to run on the gVisor runtime class

Answer: See the explanation below:

Explanation:

Install the Runtime Class for gVisor { # Step 1: Install a RuntimeClass cat <<EOF | kubectl apply -f -

apiVersion: node.k8s.io/v1beta1 kind: RuntimeClass metadata: name: gvisor handler: runsc

EOF

}

Create a Pod with the gVisor Runtime Class

{ # Step 2: Create a pod

cat <<EOF | kubectl apply -f -

apiVersion: v1

kind: Pod

metadata:

name: nginx-gvisor

spec:

runtimeClassName: gvisor

containers:

- name: nginx

image: nginx

www.atmicnetworks.com

EOF

}

Verify that the Pod is running

{ # Step 3: Get the pod

kubectl get pod nginx-gvisor -o wide

}

Question: 8

SIMULATION

You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSSHOO	kssh00301	kssh00301
301	-master	-workerl

You can switch the cluster/configuration context using the following command:

[candidate@cli] $ kubec tl config use-context KS SH00301

Task

Create a NetworkPolicy named pod-access to restrict access to Pod users-service running in namespace

dev-team.

www.atmicnetworks.com

Only allow the following Pods to connect to Pod users-service:

- Pods in the namespace qa
- Pods with label environment: testing, in any namespace

Make sure to apply the NetworkPolicy.

You can find a skeleton manifest file at

/home/candi date/KSSH00301 /n etwork'policy.yaml

Answer: See explanation below.

Explanation:

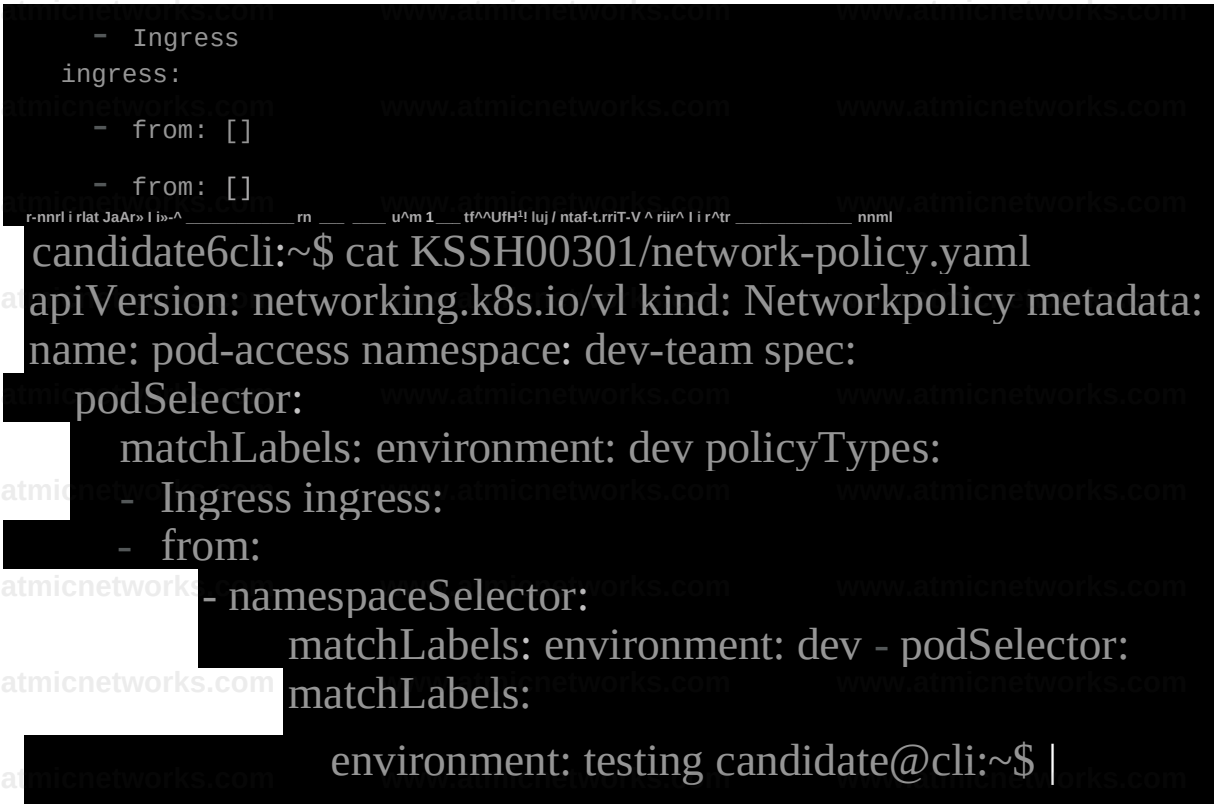
```
candidate@cli:-$ kubectl config use-context KSSH00301 Switched to context "KSSH0O301". candidate@cli:~$ candidate@cli:~$
candidate@cli:~$ kubectl get ns dev-team —show-labels NAME STATUS AGE LABELS
dev-team Active 6h39m environment=dev,kubernetes.io/metadata.name=dev~team
candidate0cli:~$ kubectl get pods -n dev-team —show-labels NAME READY STATUS RESTARTS AGE
LABELS
users-service 1/1 Running 0 6h40m environment=dev
candidate@cli:~$ Is KSCH00301 KSMV00102 KSSC0Q301 KSSHO0401 test-secret-pod.yam!
KSCS00101 KSMV00301 KSSH00301 password.txt username.txt candidate@cli:~$ vim np.yamll
```

```
networking.k8s.io/vl NetworkPolicy
metadata:
  pod-access dev-team
spec:
  podSelector:
    matchidDcis:
      dev
```

www.atmicnetworks.com

```
  policyTypes:
    Ingress
  ingress:
   - from:
     -  namespaceSelector: matchLabels:
                  ■  _ _          dev
     -  podSelector:
          matchLabeIs:
                    - testing
candidate@cli:-$ vim np.yaml
candidate0cli:-$ cat np.yaml apiVersion: networking.k8s.io/vl kind:
NetworkPolicy metadata:
  name: pod-access
  namespace: dev-team
spec:
  podselector:
    matchLabels:
      environment: dev
  policyTypes:
    -  Ingress
  ingress:
    -  from:
         namespaceSelector:
           matchLabels: environment: dev - podSelector:
           matchLabels:
             environment: testing candida te^c1i:'$ candidate@cli:~$
candidate@cli:~$ kubectl create f np.yaml n dev-team
networkpolicy.networking.k8s.io/pod-access created candidate@cli:-$ kubectl
describe netpol n dev team Name:     pod-access
Namespace:    dev-team
Created on: 2022-05-20 15:35:33 +0000 UTC Labels:      <none>
Annotations: <none>
Spec:
  PodSelector:     environment-dev
  Allowing ingress traffic:
    To Port: <any> (traffic allowed to all ports)
    From:
      NamespaceSelector: environment-dev
    From:
      PodSelector: environment=testing
  Not affecting egress traffic
  Policy Types: Ingress
candidategcli:'$ cat KSSH00301/network policy.yaml
```

```
apiVersion: networking.k8s.io/vl kind: Networkpolicy metadata:
  name: ""
  namespace: ""
spec:
  podSelector: ()
  policyTypes:
```



Question: 9

SIMULATION

A container image scanner is set up on the cluster.

Given an incomplete configuration in the directory

/etc/kubernetes/confcontrol and a functional container image scanner with HTTPS endpoint https://test-server.local.8081/image_policy

1. Enable the admission plugin.

2. Validate the control configuration and change it to implicit deny.

Finally, test the configuration by deploying the pod having the image tag as latest.

Answer: See explanation below.

Explanation:

ssh-add ~/.ssh/tempprivate

eval "$(ssh-agent -s)"

cd contrib/terraform/aws

vi terraform.tfvars

terraform init

terraform apply -var-file=credentials.tfvars

ansible-playbook -i ./inventory/hosts ./cluster.yml -e ansible_ssh_user=core -e bootstrap_os=coreos - b --become-user=root --flush-cache -e ansible_user=core

Question: 10
SIMULATION

On the Cluster worker node, enforce the prepared AppArmor profile

#include <tunables/global>

profile nginx-deny flags=(attach_disconnected) {

www.atmicnetworks.com

```
#include <abstractions/base>

file,

- Deny all file writes.
deny /** w,

}
EOF'
```

Edit the prepared manifest file to include the AppArmor profile.

```
apiVersion: v1
kind: Pod
metadata:
name: apparmor-pod
spec:
containers:
- name: apparmor-pod
  image: nginx
```

Finally, apply the manifests files and create the Pod specified on it.

Verify: Try to make a file inside the directory which is restricted.

Answer: See explanation below.

Explanation:

```
candidate$cli:~$ kubectl config use-context KSSH00401
Switched to context "KSSH00401", candidate@cli:~$ ssh kssh00401-workerl
Warning: Permanently added '10.240.86.172' (ECDSA) to the list of known hosts.
```

www.atmicnetworks.com

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

root@kssh00401-workerl:~# head /etc/apparmor.d/nginx_apparmor
#include <tunables/global>

profile nginx profile 2 flags=(attachdisconnected,mediatedeleted) (
   i include <abstractions/base>
      network inet tcp,
      network inet udp, network inet icmp,
```

```
      deny network raw,

root@kssh00401-workerl:~# apparmor_parser q /etc/apparmor.d/nginx__apparmor root@kssh00401-workerl:~4 exit
logout
Connection to 10.240.86.172 closed.
candidateScli:~$ cat KSSH00401/nginx-pod.yami

apiVersion: vl kind: Pod metadata: name: nginx-pod spec:
   containers:
   -    name: nginx-pod
        image: nginx:1.19.0
        ports:
        -    containerPort: 80
candidate@cli:~$ vim KSSH0O401/nginx-pod.yamll
```

```
          vl
            Pod

              nginx-pod
          stations:
                                  i                                    localhost/nginx-p
```

```
          : - nginx-pod
                    nginx:1.19.0
```

```
           ccmainerPort: ;
```

```
candidate@cli:'$ vim KSSH00401/nginx pod.yarn!
candidategcli:~? kubectl create -f KSSH00401/nginx pod.yaml pod/nginx-pod created
candidate6cli:~$ cat KSSH00401/nginx-pod.yaml

apiVersion: vl kind: Pod metadata:
   name: nginx-pod
   annotations:
      container, appannor.security.beta.kubernetes.io/nginx-pod: localhost/nginx-profile-2 spec: containers: - name: nginx-pod
      image: nginx:1.19.0 ports: - containerPort: 80
```

Question: 11

SIMULATION

Create a new NetworkPolicy named deny-all in the namespace testing which denies all traffic of type ingress and egress traffic

www.atmicnetworks.com

Answer: See the explanation below:

Explanation:

You can create a "default" isolation policy for a namespace by creating a NetworkPolicy that selects all pods but does not allow any ingress traffic to those pods.

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: default-deny-ingress
spec:
podSelector: {}
policyTypes:
- Ingress
```

You can create a "default" egress isolation policy for a namespace by creating a NetworkPolicy that selects all pods but does not allow any egress traffic from those pods.

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-all-egress
spec:
podSelector: {}
egress:
- {}
```

www.atmicnetworks.com

policyTypes:

- Egress

Default deny all ingress and all egress traffic

You can create a "default" policy for a namespace which prevents all ingress AND egress traffic by creating the following NetworkPolicy in that namespace.

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: default-deny-all

spec:

podSelector: {}

policyTypes:

- Ingress
- Egress

This ensures that even pods that aren't selected by any other NetworkPolicy will not be allowed ingress or egress traffic.

Question: 12

SIMULATION

a. Retrieve the content of the existing secret named default-token-xxxxx in the testing namespace.

Store the value of the token in the token.txt

b. Create a new secret named test-db-secret in the DB namespace with the following content:

username: mysql

password: password@123

Create the Pod name test-db-pod of image nginx in the namespace db that can access test-db-secret via a volume at path /etc/mysql-credentials

Answer: Answer: See the explanation below:

Explanation:

To add a Kubernetes cluster to your project, group, or instance:

Navigate to your:

Project's Operations > Kubernetes page, for a project-level cluster.

Group's Kubernetes page, for a group-level cluster.

Admin Area > Kubernetes page, for an instance-level cluster.

Click Add Kubernetes cluster.

Click the Add existing cluster tab and fill in the details:

Kubernetes cluster name (required) - The name you wish to give the cluster.

Environment scope (required) - The associated environment to this cluster.

API URL (required) - It's the URL that GitLab uses to access the Kubernetes API. Kubernetes exposes several APIs, we want the "base" URL that is common to all of them. For example, https://kubernetes.example.com rather than https://kubernetes.example.com/api/v1.

Get the API URL by running this command:

kubectl cluster-info | grep -E 'Kubernetes master|Kubernetes control plane' | awk '/http/ {print $NF}'

CA certificate (required) - A valid Kubernetes certificate is needed to authenticate to the cluster. We use the certificate created by default.

List the secrets with kubectl get secrets, and one should be named similar to default-token-xxxxx.

Copy that token name for use below.

Get the certificate by running this command:

kubectl get secret <secret name> -o jsonpath="{['data']['ca\.crt']}"

Question: 13

SIMULATION

use the Trivy to scan the following images,

1. amazonlinux:1
2. k8s.gcr.io/kube-controller-manager:v1.18.6

Look for images with HIGH or CRITICAL severity vulnerabilities and store the output of the same in /opt/trivy-vulnerable.txt

Answer: Send us your suggestion on it.

Question: 14
SIMULATION

You **must** complete this task on the following cluster/nodes:

Master node	Worker node	Cluster
KSRSOO1 ksrsOO1O1 01 -master	ksrsOO1O1- worker!	

You can switch the cluster/configuration context using the following command:

```
[ candidate^ li] $ kubec tl config use-context KS RS00101
```

You may use your browser to open **one additional tab** to access Falco's documentation.

Two tools are pre-installed on the cluster's worker node:

- sysdig
- falco

Using the tool of your choice (including any non pre-installed tool), analyze the container's behavior for at least 30 seconds, using filters that detect newly spawning and executing processes.

Store an incident file at /opt/KSRS00101/alerts/details, containing the detected incidents, one per line, in the following format:

timestamp,uid/username,proce ssName

The following example shows a properly formatted incident file:

01:40:19.601363716,root,init

01:40:20.606013716,nobody,ba sh

01:40:21.137163716,1000,tar

Keep the tool's original timestamp-format as-is

Make sure to store the incident file on the cluster's worker node.

Answer: See explanation below.

Explanation:

```
candidate@cli:~$ kubectl config use-context KSRS00101 Switched to context "KSRS00101".
candidate@cli:~$ ssh ksrsOOlOl-workerl
Warning: Permanently added '10.240.86.96' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

root@ksrs00101-workerl:~# falco falco        falco-driver-loader
root@ksrs00101-workerl:~# Is -1 /etc/falco/
```

www.atmicnetworks.com

```
total 200
-rw-r—r— 1 root root 12399 Jan 31 16:06 aws_cloudtrail_rules.yaml
-rw-r—r— 1 root root 11384 Jan 31 16:06 falco.yaml
-rw-r—r— 1 root root 1136 Jan 31 16:06 falco rules.local.yaml
-rw-r—r— 1 root root 132112 Jan 31 16:06 falco rules.yaml
-rw-r—r— 1 root root 27289 Jan 31 16:06 k8s_audit_rules.yaml
drwxr-xr-x 2 root root 4096 Feb 16 01:07 rules.available
drwxr-xr-x 2 root root 4096 Jan 31 16:28 rules.d
root@ksrs00101-workerl: ~# vim /etc/falco/falco_rules.local.yaml
```

```
i copyright (C) 2019 The Falco Authors.

I
* Licensed under the Apache License, Versi D 2.0 (the "License");
I you may not use this file except Ln compliance with the License.
I You may obtain a copy of the License at
I
J htt[ ://www.apache.• i j/1. cense; ■ LICENSE-2.0

/ Unless required by applicable law or agreed to in writing, software # distributed under the License is distributed on an “AS IS"
BASIS,
9 STTHOUT WARRANTIES OF CONDITIONS OF ANT KIND, either express r implied.
$ See the License for the specific language governing permissions and I limitations under the License.


H*+Ht4WWM4 IM
4 Your custom rules!
#r#H#t*HHHiHH(i

* Add new rules, like this one
f - rule: The program "sudo" is run in a c:trainer
I     desc: An event will trigger every time you run sudo in a container
I condition: evt.type = execve and evt.dir=< and container.id != host and proc.name = sudo # output: "Sudo run in container
(user^user.name ^container.Info parent=%proc.pname cmdli ne^^proc.cmdyjje)"
4 priority: ERROR

i tags: [users, container)
```

```
# Or override/append to any rule, macrq, or list from the Default Rules Container Drift Detected (chmod) New executable created in
                a container due to chmod
```

```
        evt.type in (open,openat,create) and

        evt.is_open_exec=               and
        container and
        not runc_writing_exec_fifo and
        not rinc_writing_var_llb_docker and

        not user_known_container_drift_activities and evt. rawres>=0
    output:

         %evt.t ime, Suser.uid,%prdc.name
          1 ERROR
```

```
root@ksrs00101-workerl:~# vim /etc/falco/falco_rules.local.yaml root@ksrs00101-workerl:'* systemctl status falco.service
#    falco.service - Falco Runtime Security

        Loaded: loaded (Zlib/systemdZsystemZfalco.service; disabled; vendor preset: enabled} Active: inactive (dead)
rootSksrsOOlOl-workerl:-# systemctl enable falco.service

Created symlink Zetc/systemd/system/multi-user.target.wants/falco.service - /lib/systemd/sys tern/falco.service.
root@ksrs00101-workerl:~f systemctl start falco.service
ropt@ksrs00101-uorkerl:~# exit
logout
Connection to 10.240.86.96 closed.
candidateOcli:-$ ssh ksrsOQlOl-workerl
last login: Fri May 20 15:59:48 2022 from 10.240.86.88
root@ksrs00101-workerl:~# vim /etc/falco/falco.yaml |

I When using json output, whether or not
```

```
#     false, the "tags" field will not be ir
```

```
I Send information logs to stderr and/ot 4 notification LOGS! These are lust Fait logstderr: true
```

```
/opt/KSRSOOlOl/alerts/details
```

```
It log level of falco1 s internal logging.
J "alert", "critical", "error", "warning : info
```

```
root@ksrs00101-worker]:~t vim /etc/falco/falco.yaml

root@ksrs00101-workerl:'l grep log /etc/falco/falco.yaml
i cloudtrail log files.

f If true, the times displayed in log messages and output messages

* Send information logs to stderr and/or syslog Note these are *not* security
#    notification t /s! These are just Falco lifecycle (and possibly error J s. logstderr: true log syslog: true
log_fHe: /opt/KSRS00101/alerts/details

I Minimum log level to include in logs* Note: these levels are

* 1 : level of falco's internal 1 jging. Can be one of "emergency", loglevel: info
#     - . f: log a DEBUG message noting that the buffer was full

#    Notice it is not possible to ignore and 1 [/alert messages at the same time.
I The rate at which log/alert messages are emitted is governed by a - log
#    The timeout error will be reported to the log according to the above log_* settings. sysiog_output:

4     - logging (alternate method than sys ■) :

#                 program: logger -t falco-test
#   this information will be ged, however the main Falco daemon will not be stopped. root@ksrs00101 worker!:-# systemctl
restart falco.service root@ksrs00101-worker1:'# exit logout Connection to 10.240.36.36 closed.
candidate@cli:~$ |
```

Question: 15

SIMULATION

Create a User named john, create the CSR Request, fetch the certificate of the user after approving it.

Create a Role name john-role to list secrets, pods in namespace john

Finally, Create a RoleBinding named john-role-binding to attach the newly created role john-role to the user john in the namespace john.

To Verify: Use the kubectl auth CLI command to verify the permissions.

www.atmicnetworks.com

Answer: See the Explanation below.

Explanation:

se kubectl to create a CSR and approve it.

Get the list of CSRs:

kubectl get csr

Approve the CSR:

kubectl certificate approve myuser

Get the certificate

Retrieve the certificate from the CSR:

kubectl get csr/myuser -o yaml

here are the role and role-binding to give john permission to create NEW_CRD resource:

kubectl apply -f roleBindingJohn.yaml --as=john

rolebinding.rbac.authorization.k8s.io/john_external-rosource-rb created

kind: RoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: john_crd

namespace: development-john

subjects:

- kind: User

name: john

apiGroup: rbac.authorization.k8s.io

roleRef:

kind: ClusterRole

www.atmicnetworks.com

name: crd-creation

kind: ClusterRole

apiVersion: rbac.authorization.k8s.io/v1 metadata:

name: crd-creation

rules:

- apiGroups: ["kubernetes-client.io/v1"] resources: ["NEW_CRD"] verbs: ["create, list, get"]

Question: 16

SIMULATION

Fix all issues via configuration and restart the affected components to ensure the new setting takes effect.

Fix all of the following violations that were found against the API server:-

a. Ensure that the RotateKubeletServerCertificate argument is set to true.

b. Ensure that the admission control plugin PodSecurityPolicy is set.

c. Ensure that the --kubelet-certificate-authority argument is set as appropriate.

Fix all of the following violations that were found against the Kubelet:-

a. Ensure the --anonymous-auth argument is set to false.

b. Ensure that the --authorization-mode argument is set to Webhook.

Fix all of the following violations that were found against the ETCD:-

a. Ensure that the --auto-tls argument is not set to true

b. Ensure that the --peer-auto-tls argument is not set to true

Hint: Take the use of Tool Kube-Bench

www.atmicnetworks.com

Answer: See the Explanation below.

Explanation:

Fix all of the following violations that were found against the API server:-

a. Ensure that the RotateKubeletServerCertificate argument is set to true.

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

component: kubelet

tier: control-plane

name: kubelet

namespace: kube-system

spec:

containers:

- command:

- kube-controller-manager

+ - --feature-gates=RotateKubeletServerCertificate=true

image: gcr.io/google_containers/kubelet-amd64:v1.6.0

livenessProbe:

failureThreshold: 8

httpGet:

host: 127.0.0.1

www.atmicnetworks.com

```
      path: /healthz
      port: 6443
      scheme: HTTPS
    initialDelaySeconds: 15
    timeoutSeconds: 15
  name: kubelet
  resources:
    requests:
      cpu: 250m
  volumeMounts:
  - mountPath: /etc/kubernetes/
    name: k8s
    readOnly: true
  - mountPath: /etc/ssl/certs
    name: certs
  - mountPath: /etc/pki
    name: pki
hostNetwork: true
volumes:
- hostPath:
    path: /etc/kubernetes
  name: k8s
- hostPath:
    path: /etc/ssl/certs
  name: certs
- hostPath:
```

www.atmicnetworks.com

```
      path: /etc/pki
    name: pki
```

b. Ensure that the admission control plugin PodSecurityPolicy is set.

```
audit: "/bin/ps -ef | grep $apiserverbin | grep -v grep"
      tests:
       test_items:
        - flag: "--enable-admission-plugins"
          compare:
            op: has
            value: "PodSecurityPolicy"
          set: true
       remediation: |
         Follow the documentation and create Pod Security Policy objects as per your environment.
         Then, edit the API server pod specification file $apiserverconf
         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.
       scored: true
```

c. Ensure that the --kubelet-certificate-authority argument is set as appropriate.

www.atmicnetworks.com

```
audit: "/bin/ps -
ef | grep
$apiserverbin |
grep -v grep"
                tests:
                  test_items:
                  - flag: "--kubelet-certificate-authority"
                    set: true
                remediation: |
                  Follow the Kubernetes documentation and setup the TLS connection between the
                  apiserver and kubelets. Then, edit the API server pod specification file
                  $apiserverconf 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>
                scored: true
```

Fix all of the following violations that were found against the ETCD:-

a. Ensure that the --auto-tls argument is not set to true

Edit the etcd pod specification file $etcdconf on the master node and either remove the --auto-tls parameter or set it to false. --auto-tls=false

b. Ensure that the --peer-auto-tls argument is not set to true

Edit the etcd pod specification file $etcdconf on the master
node and either remove the --peer-auto-tls parameter or set it to false.
--peer-auto-tls=false

www.atmicnetworks.com

Question: 17

SIMULATION

Create a PSP that will only allow the persistentvolumeclaim as the volume type in the namespace restricted.

Create a new PodSecurityPolicy named prevent-volume-policy which prevents the pods which is having different volumes mount apart from persistentvolumeclaim.

Create a new ServiceAccount named psp-sa in the namespace restricted.

Create a new ClusterRole named psp-role, which uses the newly created Pod Security Policy preventvolume-policy

Create a new ClusterRoleBinding named psp-role-binding, which binds the created ClusterRole psp- role to the created SA psp-sa.

Hint:

Also, Check the Configuration is working or not by trying to Mount a Secret in the pod maifest, it should get failed.

POD Manifest:

apiVersion: v1

kind: Pod

metadata:

name:

spec:

containers:

- name:

image:

volumeMounts:

- name:

mountPath:

volumes:

- name:

secret:

www.atmicnetworks.com

secretName:

Answer: See the Explanation below:

Explanation:

apiVersion: policy/v1beta1

kind: PodSecurityPolicy

metadata:

name: restricted

annotations:

seccomp.security.alpha.kubernetes.io/allowedProfileNames: 'docker/default,runtime/default'

apparmor.security.beta.kubernetes.io/allowedProfileNames: 'runtime/default'

seccomp.security.alpha.kubernetes.io/defaultProfileName: 'runtime/default'

apparmor.security.beta.kubernetes.io/defaultProfileName: 'runtime/default'

spec:

privileged: false

- Required to prevent escalations to root.

allowPrivilegeEscalation: false

- This is redundant with non-root + disallow privilege escalation,

- but we can provide it for defense in depth.

requiredDropCapabilities:

- ALL

- Allow core volume types.

volumes:

- 'configMap'

- 'emptyDir'

www.atmicnetworks.com

```
  - 'projected'
  - 'secret'
  - 'downwardAPI'
  - Assume that persistentVolumes set up by the cluster admin are safe to use.
  - 'persistentVolumeClaim'
hostNetwork: false
hostIPC: false
hostPID: false
runAsUser:
  - Require the container to run without root privileges.
  rule: 'MustRunAsNonRoot'
seLinux:
  - This policy assumes the nodes are using AppArmor rather than SELinux.
  rule: 'RunAsAny'
supplementalGroups:
  rule: 'MustRunAs'
  ranges:
    - Forbid adding the root group.
    - min: 1
      max: 65535
fsGroup:
  rule: 'MustRunAs'
  ranges:
    - Forbid adding the root group.
    - min: 1
      max: 65535
```

www.atmicnetworks.com

readOnlyRootFilesystem: false

Question: 18
SIMULATION

Given an existing Pod named nginx-pod running in the namespace test-system, fetch the serviceaccount-name used and put the content in /candidate/KSC00124.txt

Create a new Role named dev-test-role in the namespace test-system, which can perform update operations, on resources of type namespaces.

Create a new RoleBinding named dev-test-role-binding, which binds the newly created Role to the Pod's ServiceAccount (found in the Nginx pod running in namespace test-system).

Answer: See explanation below.

Explanation:

```
candidate@cli:~$ kubectl config use-context KSCH00201 Switched to context "KSCH00201". candidategcii:~$ kubectl get pods -n
security NAME          READY       STATUS       RESTARTS          AGE
web-pod      1/1          Running       0                6h9m
candidate@cli:~$ kubectl get deployments.apps -n security No resources found in security namespace.
candidateScli:~$ kubectl describe rolebindings.rbac.authorization.kBs.io -n security Name:      dev-role
labels:              <none>
Anhota t ions: <none> Role:
    Kind: Role
    Name: dev-role
Subjects:
    Kind                    Name Namespace

    ServiceAccount sa-dev-1
candidate?cli:~$ kubectl describe role dev-role n security
Name:             dev-role
Labels:           <none>
Annotations: <none>
PolicyRule:
    Resources Non-Resource URLs Resource Names Verbs

    *          [J                  []                  [*]
candidate@cli:~$ kubectl edit role/dev-role -n security |
```

www.atmicnetworks.com

b4c9ddd6-2729-43bd-8fbd-b2d227f4c4cd

iGroupsi

services

- watch

```
candidate9cli:~$ kubectl describe role dev-role -n security
Name:         dev-role
Labels:       <none>
Annotations: <none>
PolicyRule:
  Resources Non-Resource URLs Resource Names Verbs

    *          [1                [1            [*l
candidate^cli:-$ kubectl edit role/dev-role -n security role.rbac.authorization.kBs.io/dev-role edited candidateScli:-? kubectl
describe role dev-role -n security Name:.  dev-role
Labels:       <nohe>
Annotations: <none> PolicyRule: Resources Non-Resource URLs Resource Names Verbs services []      []          [watch]

candidateScli:~5 kubectl get pods -h security NAME            READY   STATUS  RESTARTS          AGE
web-pod     1/1          Running     0               6hl2m
candidate@cli:-$ kubectl get pods/web-pod n Security -o yaml ( grep serviceAccount serviceAccount: sa-dev-1 servi
:eAccountjfame: sa-dev-1 - serviceAccountTOkeni
candidate@cli:~S kubectl create role role-2 --verb-update —resource-namespaces -n security role.rbac.authorization.kBs.io/role-2
created candidate@cli:'$ kubectl create rolebinding role-2-binding —role —role      —role=
candidate@cli:~5 kubectl create rolebinding role-2-binding - role=role-2 --serviceaccount=se curity:sa-dev-1 -n security
rolebinding.rbac.authorization. kBs.io/role-2-binding created candidatc@cli:~$ Q
```

Question: 19

SIMULATION

Enable audit logs in the cluster, To Do so, enable the log backend, and ensure that

1. logs are stored at /var/log/kubernetes/kubernetes-logs.txt.
2. Log files are retained for 5 days.
3. at maximum, a number of 10 old audit logs files are retained.

Edit and extend the basic policy to log:

1. Cronjobs changes at RequestResponse
2. Log the request body of deployments changes in the namespace kube-system.
3. Log all other resources in core and extensions at the Request level.
4. Don't log watch requests by the "system:kube-proxy" on endpoints or

Answer: See explanation below.

Explanation:

```
candidateScli:^ kubectl config use-context KSRS00602 Switched to context "KSRS0O602".
candidate? Cli :^ ssh ksrs00602-master

Warning: Permanently added '10.240.36.243' (ECDSA) to the list of known hosts.
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

root@ksrs00602-master:~l cat /etc/kubernetes/logpolicy/sample-policy.yaml
```

```
apiVersion: audit.kBs.io/vl kind: Policy
# Don't generate audit events for all requests in RequestReceived stage.

omitStages:
    -    "RequestReceived"

rules:
    I Don't log watch requests by the "system:kube-proxy" on endpoints or services - level: None
      users: ["system:kube-proxy"] verbs: ["watch"] resources:
      -  group: ”“ # core API group resources: ["endpoints", "services"]
```

```
    -    Don’t log authenticated requests to certain non-resource URL paths.

    -  level: None userGroups: ["system:authenticated"] nonResourceURLs:
       -   "/api*" # Wildcard matching.

       -    "/version"
    # Edit form here below
root@ksrs00602-master:^i vim /etc/kubernetes/logpolicy/sample-policy.yaml
```

www.atmicnetworks.com

```
    - "/api*" # Wildcard matching.
    - "/version"
  # Edit form here below
  - level: RequestResponse
    resources:
    - group: ""
      resources: ["cronjobs"]
  - level: Request
    resources:
    - group: "" # core API group
      resources: ["pods"]
      namespaces: ["webapps"]
# Log configmap and secret changes in all other namespaces at the Metadata level.
  - level: Metadata
    resources:
    - group: "" # core API group
      resources: ["secrets", "configmaps"]

  # A catch-all rule to log all other requests at the Metadata level.
  - level: Metadata
    # Long-running requests like watches that fall under this rule will not
    # generate an audit event in RequestReceived.
    omitStages:
      - "RequestReceived"
```

```
      - "/version"
  -   Edit form here below
  -   level.: RequestRespoiise
      resources:

  -   group: "" resources: ["cronjobs"]
  -   level: Request
      resources:
  -   group: "" 1 core API group resources: ["pods"] namespaces: ["webapps"]
S Log configmap and secret changes in all other namespaces at the Metadata level. - level: Metadata
      resources:

  -   group: "" J core API group
      resources: ["secrets", "configmaps"]

f A catch-all rule to log all other requests at the Metadata level.
  -   level: Metadata
      t Long-running requests like watches that fall under this rule will not
  -   generate an audit event in RequestReceived.
      omitStages:
        -   "RequestReceived"

root@ksrs00602-master:~i vim /etc/kubemetes/logpolicy/sample-policy.yaml root@ksrs00602-master:~i vim
/etc/kubernetes/manifests/kube-apiserver.yaml |

                kube-apiserver control-plane kube-apiserver
              kube-system
```

```
              kube-apiserver
              —advert ise-address=10.240.86.243
          -   —allow-privileged=
          -   —audit-policy-file=/etc/kubernetes/logpolicy/sample-policy.yaml --audit-log-
              path=/var/log/kubernetes/kubernetes-logs.txt --audit-log-maxbackup=l
          ----- audit-log-maxage-30
```

```
-   —authorization-mode^Node,RBAC
    —client-ca-file=/etc/kubernetes/pki/ca.crt
    --enable-admission-plugins=NodeRestriction   enable-bootstrap-token-auth=
    —etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
```

```
I A catch-all rule to log all other requests at the Metadata level.
  -  level: Metadata
    -   Long-running requests like watches that fall under this rule will not # generate an audit event in RequestReceived.
      omitStages:
    -   "RequestReceived"
root@ksrs00602-master:-I vim /etc/kubernetes/logpolicy/sample-policy.yaml root@ksrs00602 master:-I vim /etc/kubernetes/manifests/kube-apiserver.yaml root@ksrs00602-master:-i systemctl daemon-reload root@ksrs00602-master:~f systemctl restart kubelet.service root@ksrs00602 master:-! systemctl enable kubelet root@ksrs00602-master:-! exit logout
Connection to 10.240.86.243 closed.
candidate@cli:-$ I
```

Question: 20

SIMULATION

Create a RuntimeClass named untrusted using the prepared runtime handler named runsc.

Create a Pods of image alpine:3.13.2 in the Namespace default to run on the gVisor runtime class.

Answer: See the explanation below:

Explanation:

```
Starting gVisor...
creating cloned children...
Moving files to filing cabinet..
Letting the watchdogs out...
Digging up root...
Gathering forks...
Daemonizing children...
Rewriting operating system in Javascript...
Reading process obituaries...
Waiting for children...
Segmenting fault lines...
Ready!
```

www.atmicnetworks.com

Question: 21
SIMULATION

Create a network policy named allow-np, that allows pod in the namespace staging to connect to port 80 of other pods in the same namespace.

Ensure that Network Policy:-

1. Does not allow access to pod not listening on port 80.
2. Does not allow access from Pods, not in namespace staging.

Answer: See the explanation below:

Explanation:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: network-policy
spec:
podSelector: {} #selects all the pods in the namespace deployed policyTypes:
- Ingress
ingress:
- ports:     #in input traffic allowed only through 80 port only
  - protocol: TCP
    port: 80
```

Question: 22
SIMULATION

Create a Pod name Nginx-pod inside the namespace testing, Create a service for the Nginx-pod named nginx-svc, using the ingress of your choice, run the ingress on tls, secure port.

Answer: See explanation below.

Explanation:

$ kubectl get ing -n <namespace-of-ingress-resource>

NAME HOSTS ADDRESS PORTS AGE

cafe-ingress cafe.com 10.0.2.15 80 25s

$ kubectl describe ing <ingress-resource-name> -n <namespace-of-ingress-resource>

Name: cafe-ingress

Namespace: default

Address: 10.0.2.15

Default backend: default-http-backend:80 (172.17.0.5:8080)

Rules:

Host Path Backends

cafe.com

/tea tea-svc:80 (<none>)

/coffee coffee-svc:80 (<none>)

Annotations:

kubectl.kubernetes.io/last-applied-configuration:
{"apiVersion":"networking.k8s.io/v1","kind":"Ingress","metadata":{"annotations":{},"name":"cafe-ingress","namespace":"default","selfLink":"/apis/networking/v1/namespaces/default/ingresses/cafe -ingress"},"spec":{"rules":[{"host":"cafe.com","http":{"paths":[{"backend":{"serviceName":"tea-svc","servicePort":80},"path":"/tea"},{"backend":{"serviceName":"coffee-svc","servicePort":80},"path":"/coffee"}]}}]},"status":{"loadBalancer":{"ingress":[{"ip":"169.48.142.11 0"}]}}}

www.atmicnetworks.com

Events:

Type Reason Age From Message

Normal CREATE 1m ingress-nginx-controller Ingress default/cafe-ingress

Normal UPDATE 58s ingress-nginx-controller Ingress default/cafe-ingress

$ kubectl get pods -n <namespace-of-ingress-controller>

NAME READY STATUS RESTARTS AGE

ingress-nginx-controller-67956bf89d-fv58j 1/1 Running 0 1m

$ kubectl logs -n <namespace> ingress-nginx-controller-67956bf89d-fv58j

NGINX Ingress controller

Release: 0.14.0

Build: git-734361d

Repository: https://github.com/kubernetes/ingress-nginx

Question: 23

SIMULATION

Secrets stored in the etcd is not secure at rest, you can use the etcdctl command utility to find the secret value

for e.g:-

ETCDCTL_API=3 etcdctl get /registry/secrets/default/cks-secret --cacert="ca.crt" --cert="server.crt" --key="server.key"

Output

Using the Encryption Configuration, Create the manifest, which secures the resource secrets using the provider AES-CBC and identity, to encrypt the secret-data at rest and ensure all secrets are encrypted with the new configuration.

Answer: See explanation below.

Explanation:

ETCD secret encryption can be verified with the help of etcdctl command line utility.

ETCD secrets are stored at the path /registry/secrets/$namespace/$secret on the master node.

The below command can be used to verify if the particular ETCD secret is encrypted or not.

- ETCDCTL_API=3 etcdctl get /registry/secrets/default/secret1 [...] | hexdump -C

Question: 24
SIMULATION

Service is running on port 389 inside the system, find the process-id of the process, and stores the names of all the open-files inside the /candidate/KH77539/files.txt, and also delete the binary.

Answer: See explanation below.

Explanation:

www.atmicnetworks.com

```
root# netstat -ltnup

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address           Foreign Address State PID/Program name
tcp       0      0 127.0.0.1:17600     0.0.0.0:*           LISTEN 1293/dropbox
tcp       0      0 127.0.0.1:17603     0.0.0.0:*           LISTEN 1293/dropbox
tcp       0      0 0.0.0.0:22          0.0.0.0:*           LISTEN 575/sshd
tcp       0      0 127.0.0.1:9393      0.0.0.0:*           LISTEN 900/perl
tcp       0      0 :::80               :::*                LISTEN 9583/docker-proxy
tcp       0      0 :::443              :::*                LISTEN 9571/docker-proxy
udp 0          0 0.0.0.0:68          0.0.0.0:*                   8822/dhcpcd


root# netstat -ltnup | grep ':22'
tcp 0          0 0.0.0.0:22          0.0.0.0:*           LISTEN 575/sshd
```

The ss command is the replacement of the netstat command.

Now let's see how to use the ss command to see which process is listening on port 22:

```
root# ss -ltnup 'sport = :22'
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
tcp LISTEN 0      128    0.0.0.0:22        0.0.0.0:*        users:("sshd",pid=575,fd=3))
```

Question: 25

SIMULATION

Use the kubesec docker images to scan the given YAML manifest, edit and apply the advised changes, and passed with a score of 4 points.

kubesec-test.yaml

```
apiVersion: v1
```

www.atmicnetworks.com

kind: Pod

metadata:

name: kubesec-demo

spec:

containers:

- name: kubesec-demo

image: gcr.io/google-samples/node-hello:1.0

securityContext:

readOnlyRootFilesystem: true

Hint: docker run -i kubesec/kubesec:512c5e0 scan /dev/stdin < kubesec-test.yaml

Answer: See explanation below.

Explanation:

kubesec scan k8s-deployment.yaml

cat <<EOF > kubesec-test.yaml

apiVersion: v1

kind: Pod

metadata:

name: kubesec-demo

spec:

containers:

- name: kubesec-demo

image: gcr.io/google-samples/node-hello:1.0

securityContext:

readOnlyRootFilesystem: true

www.atmicnetworks.com

```
EOF

kubesec scan kubesec-test.yaml

docker run -i kubesec/kubesec:512c5e0 scan /dev/stdin < kubesec-test.yaml

kubesec http 8080 &

[1] 12345
{"severity":"info","timestamp":"2019-05-
12T11:58:34.662+0100","caller":"server/server.go:69","message":"Starting HTTP server on port 8080"}

curl -sSX POST --data-binary @test/asset/score-0-cap-sys-admin.yml http://localhost:8080/scan

[
{
  "object": "Pod/security-context-demo.default",
  "valid": true,
  "message": "Failed with a score of -30 points",
  "score": -30,
  "scoring": {
    "critical": [
      {
        "selector": "containers[] .securityContext .capabilities .add == SYS_ADMIN",
        "reason": "CAP_SYS_ADMIN is the most privileged capability and should always be avoided"
      },
      {
        "selector": "containers[] .securityContext .runAsNonRoot == true",
        "reason": "Force the running image to run as a non-root user to ensure least privilege"
```

www.atmicnetworks.com

```
    },
// ...
```

Question: 26
SIMULATION

Using the runtime detection tool Falco, Analyse the container behavior for at least 20 seconds, using filters that detect newly spawning and executing processes in a single container of Nginx.

store the incident file art /opt/falco-incident.txt, containing the detected incidents. one per line, in the format

[timestamp],[uid],[processName]

Answer: Send us your feedback on it.

Question: 27
SIMULATION

Cluster: qa-cluster
Master node: master Worker node: worker1

You can switch the cluster/configuration context using the following command: [desk@cli] $ kubectl config use-context qa-cluster

Task:
Create a NetworkPolicy named restricted-policy to restrict access to Pod product running in namespace dev.

Only allow the following Pods to connect to Pod products-service:
1. Pods in the namespace qa
2. Pods with label environment: stage, in any namespace

Answer: See the Explanation below.

Explanation:

```
candidate?cli:~$ kubectl config use-context KSSH003O1 Switched to context "KSSH00301". candidate?cli:~$ candidateScli: ~$
```

www.atmicnetworks.com

```
candidate@cli;~$ kubectl get ns dev-team —show-labels NAME          STATUS  AGE       LABELS
dev-team       Active       6h39m       environment=dev,kubernetes.io/metadata.name=dev-team
candidate9cli:~? kubectl get pods -n dev-team --show-labels NAME          READY STATUS  RESTARTS          AGE
                    LABELS
users-service          1/1          Running       0                6h40m       environment=dev
candidateScli:~$ Is
KSCHO03O1 KSMV00102 KSSC00301 KSSH00401                test-secret-pod.yaml
KSCSO0101 KSMV00301 KSSH003O1 password.txt username.txt candidateScli:~$ vim np.yaml|
```

```
                networking.k8s.io/vl NetworkPolicy
metadata:
          pod-access dev-team
spec:
  podSelector:
    matchLabels:
                dev policyTypes:
Ingress
  Ingress:
    - from:
                - namespaceSelector:
                      matchLabels:
              . _i              dev
          -  podSelector: matchLabels:
                      -i testing
```

www.atmicnetworks.com

```
candidate@cli:-$ vim np.yaml
candidate0cli:-$ cat np.yaml apiVersion: networking.k8s.io/vl kind:
NetworkPolicy metadata:
  name: pod-access
  namespace: dev-team
spec:
  podselector:
    matchLabels:
      environment: dev
  policyTypes:
  - Ingress
  ingress:
  - from:
      namespaceSelector:
        matchLabels: environment: dev - podSelector:
      matchLabels:
        environment: testing candida te^c1i:'$ candidate@cli:~$
candidate@cli:~$ kubectl create f np.yaml n dev-team
networkpolicy.networking.k8s.io/pod-access created candidate@cli:-$ kubectl
describe netpol n dev team Name:    pod-access
Namespace:   dev-team
Created on: 2022-05-20 15:35:33 +0000 UTC Labels:      <none>
Annotations: <none>
Spec:
  PodSelector:     environment-dev
  Allowing ingress traffic:
    To Port: <any> (traffic allowed to all ports)
    From:
      NamespaceSelector: environment-dev
    From:
      PodSelector: environment=testing
  Not affecting egress traffic
  Policy Types: Ingress
candidateBcli:-$ cat KSSH00301/network policy.yaml
```

```
apiVersion: networking.k8s.io/vl kind: Networkpolicy metadata:
  name: ""
  namespace: ""
spec:
  podSelector: ()
  policyTypes:
  - Ingress
  ingress:
  - from: []
  - from: []
```

r-nnrl i rlat ja Ar» I i»-^ _____ rn ___ ___ u^m 1___ lfg^UfU^ hl I /n«tt.rnrV-.rin 1 1 ru _____ irnm I

candidate@cli:~$ cat KSSH00301/network policy.yaml apiVersion: networking.k8s.io/vl kind: Networkpolicy metadata: name: pod-access namespace: dev-team spec: podSelector: matchLabels: environment: dev policyTypes:

```
- Ingress ingress:
- from:
  - namespaceSelector: matchLabels: environment: dev
  - podSelector:
      matchLabels:
        environment: testing candidate@cli:~$ |
```

Reference: https://kubernetes.io/docs/concepts/services-networking/network-policies/

Question: 28
SIMULATION

You must complete this task on the following duster/nod

Cluster Master node Worker node

switch the cluster/configuration context using the

Context

A container image scanner is set up on the cluster, but it's not yet fully integrated into the cluster s configuration. When complete, the container image scanner shall scan for and reject the use of vulnerable images.

Task

www.atmicnetworks.com

You have to complete the entire task on the cluster's master node where all services and files have been prepared and placed.

Given an incomplete configuration in directory /etc/kubernetes/epconfig and a functional container image scanner with HTTPS endpoint https://wakanda.local:8081 /image_policy :

1. Enable the necessary plugins to create an image policy
2. Validate the control configuration and change it to an implicit deny
3. Edit the configuration to point to the provided HTTPS endpoint correctly

Finally, test if the configuration is working by trying to deploy the vulnerable resource /root/KSSC00202/vulnerable-resource.yml.

You can find the container image scanner's log file at /var/log/imagepolicy/acme.log

Answer: See the explanation below

Explanation:

```
Switched to context "KSSC6G202".
candidoteQcli;-4 ssh kSSCSB262-master
Warning: Permanently added 10.177.80.12' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the individual files in /usr/share/doc/'/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

root^ks5C0O292-master;-# Is /etC/kubernetes/epconfig/
admission configuration.json apiserver-client key.pem apiserver*client.pen kubeconfig.yaml webhook-key.pern webhook.pern
root^kssc00202-master;-W vim /etc/kubernetes/epconfig/admission configuration.json |
```

```
" imagePolicy": {
  "kubeConfigFile": "/etc/kubernetes/epconfig/kubeconfig.yaml",
  "allowTTL": 58, "denyTTL": 50, "retry8ackoff": 500, "defaultAt low":
```

www.atmicnetworks.com

```
false!
```

```
root$kssc00202-master:-# vim /etc/kubernetes/epconfig/admission configuration.json rootakssc00202-master:-# vim /etc/kubernetes/epconfig/admission configuration.json root(akssc002O2 master:-# vim /etc/kubernetes/epconfig/kubeconfig.yaml|
apiVersion: vl clusters: - cluster:
    certificate-authority: /etc/kubernetes/epconfig/webhook.pem * CA for verifying the renote service, server: https://wakanda.local:8981/image policy!
  name: kubernetes contexts: ■ context: cluster: kubernetes user: kubernetes-adfflin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin^kubernetes
kind: Config
preferences: {} users: • name: kubernetes-admin user:
    client-certificate: /etc/kubernetes/epconfig/apiserver-client.pen
    client-key: /etc/kubernetes/epconfig/apiserver-client•key.pern
```

```
root^kssc00202-master:~# vim /etc/kubernetes/epconfxg/admxssion configuration.json root?kssc00202-master:-# vim /etc/kubernetes/epconfig/admission configuration.json root@k$$c00202•master:-# vim /etc/kubernetes/epconfig/kubeconfig.yaml root^kssc0O2O2 master:-# vim /etc/kubernetes/manifests/kube apiserver.yaml p|
```

```
EpiVersion; vl kind: Pod metadata: annotations: kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.177.80.12:6443 creationTimestamp: null labels: component: kube-apiserver tier: control-plane name: kube apiserver namespace: kube-system spec: containers: • command: • kube-apiserver - - advertise-address=10.177.80.12 - --allow-privileged=true - —authorization-modeMtodeaRBAC - --client-ca-file=/etc/kubernetes/pki/ca.crt • - enable admission plugins-NodeRestriction • --enable-bootstrap-token auth=true - •-etcd-caf ile»/etc/kubemetes/pki/etcd/ca.crt
    - etcd - cert file«/et c/kubernetes/pki/apiserver etcd- cllent.crt - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key - - etcd servers-https://127.0.0.1:2379 - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt - - kubelet-client-key»/etc/kube met es/pki/apiserver-kubelet•client.key • --kubelet-preferred- address-types-InternalIP,ExternalIP.Hostname - • proxy-cllent-cert-file»/etc/kubernetes/pkl/front proxy-cllent.crt - - proxy-client key flle'/etc/kubernetes/pkl/front proxy-client.key - ••requestheader-allowed-names"front-proxy-client - - - requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt - --request heade r-ext ra heade rs-p ref i x»X-Remot e E x t ra
"/etc/kubernetes/Banifests/kube-apiserver.yaml* 135L, 4626C
root?kssc00202-master:-# vim /etc/kubernetes/manifests/kube-apiserver.yaml p 2 files to edit
root^kssc00202master:-# rm -f p
root@ks5C00202-master:-# vim /etc/kubernetes/manjfests/kube apiserver.yaml |
```

```
apiVers ion: vl kind: Pod metadata: annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.177.86.12:6443 creation!imestamp: null
  labels:
    component: kube-aplserver tier: control-plane
  name: kube-aplserver
  namespace: kube-system spec:
  containers:
    • command:
      • kube-aplserver
      • -advertise address=10 177.86.12
      • •-allow-privileged=true
      • ••authorization-mode=Node,RBAC
      • •-client-ca-file=/etc/kubernetes/pki/ca.crt
      • enable admission plugms-NodeRestriction,ImagePollcyWebHoofl
      • • admission•cont rol- config • file»/etc/kube rnetes/epconfig/adnun.con f
      • - enable-bootstrap-token-auth»true
      • •-etcd-cafile-/etc/kubernetes/pki/etcd/ca.crt
      • --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-cl lent.crt
```

www.atmicnetworks.com

- - etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
- •-etcd-servers=https://127.0.0.1:2379
- - - kubelet - cl lent-cert if icate<»/et c/kubernetes/pki/apiserver-kubelet-cl lent .crt • --kubelet-client key-Zetc/kubernetes/pki/apiserver-kubelet client.key
- - kubelet•preferred-address-types-InternalIP,External IP,Hostname - proxy-client cert-file-ZetcZkubernetesZpkiZfront-proxy-client.crt - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key - - -requestheader-allowed-names=front-proxy-client
- --requestheader-cllent-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt

```
rootakssc0O2O2-master:-« rm -f p
root@kssc0O2O2•master:-# vim Zetc/kubernetes/manifests/kube•apiserver.yaml root^kssc00202
master:-# systemctl daemon-reload root^kssc00202•master:-#
root@kssc0O2O2- master:-# root^kssc00202-master:-# root<3kssc0O202 master:-#
root@kssc0O2O2 master:-# systemctl restart kubelet.service root@kssc00202-master:~# systemctl
enable kubelet.service rootdkssc0O2O2 master:-#
rootakssc©0202-mast er:-#
root3kssc002O2-master:-#
root?kssc0O2O2-master:-# roottakssc00202-master;-# Is
KSSC00202 snap
root@kssc0O202-master:-# cat KSSC00202/vulnerable-resource.yml |
KSSC00202 snap
root£ks$c60202-master: # cat KSSC00202/vulnerable resource.yml

apiVersion: vl
kind: Replicationcontroller
metadata:
   name: nginx-latest
spec:

   replicas: 1
   selector:

     app: nginx-latest template:
     metadata:
       name: nginx-latest labels:
           app: nginx-latest spec:
         containers:

       - name: nginx-latest image: nginx ports:
         - containerPort: 80

root(dkssc002O2 master:-# kubectl create f KSSCO0202/vulnerable-resource.yml|
```

```
root®ksscO02O2 master:-# kubectl create -f K$SC002O2/vulnerable-resource.yml
The connection to the server 10.177.80.12:6443 was refused - did you specify the right host or port7 root@kssc002O2
master:-# kubectl get pods
The connection to the server 10.177.80.12:6443 was refused ■ did you specify the right host or port? root^kssc00202
master:-# Is -al .kube/ total 20 drwxr-xr-x3 root     root          4096       Aug        3            04:07      ,
drwx..... .... 9  root   root  4096  Oct 11 15:36    ..
drwxr-x---       4    rootroot  4096 Aug   3 04:07    cache
-rw-r--r--       1    rootroot  5636 Aug   3 04:07    config
root(akssc002O2 master:-# crictl ps -a
|
```

```
G12ea8587138e          a634548d 18MJ3          2 months ago              Exited                  kube     proxy                  0                      1468a9f
aOfleG          kube proxy cmjbS
485227bfa49d(»          aebe758cef4cd          2 months ago              Exited                  etcd                            8                      cfb6522
e720«b          etcd-kssc O02O2 master
r oot ;dkssc00202-master : ~# Is -al .kube/ | grep kube opi root4ks5c0O2O2-master:-# crictl ps -a [ grep kube-api WARN[G&0G| runtime connect using default endpoints;
|unix:///var/run/docker$hin.sock unix:///run/containerd/containerd.sock unix:///ru n/crio/crio.sock unix:///var/run/cri dockerd.sock|, As the default settings are now deprecated, you should
set the endpoint instead.
          [0000] unable to determine runtime API version: rpc error: code = Unavailable desc = connection error: desc = "transport: Error wh lie dialing dial unix /var/run/dockershin.soclc
connect: no such file or directory* HAHri[00G0| image connect using default endpoints: |unix:///var/run/dockershlm.sock unix:///run/conlalnerd/containerd.SOCk unix:///run/
cnoZcrio.sock unix:///var/run/cri-dockerd.sock). As the default settings are now deprecated, you should set the endpoint instead.
          [80001 unable to determine image API version: rpc error: code = Unavailable desc ■ connection error: desc » "transport: Error whil e dialing dial unix /varZrun/dockershim.sock:
connect: no such file or directory* a0O3b3fdfb61c d3377ffb7177c          38 seconds ago            Exited                  kube-apiserver          3 2dadb4c
984o91          kube-apiserver-ks5cGG262 master
5e7Gb9a7Gf9ed          d3377ffb7177c          7 hours ago               Exited                  kube-apiserver          8                      68a9f31
6c2559          kube-apiserver-ksscGG202 master
root ?ksSC00202-master:-# rootak5sc902O2-master:-A raot£kssc88282-master:-4 rooL£k5sc00202-master:~# exit logout Connection to 10.177.88.12 closed. candidate@cli:-$ |
```

Question: 29
SIMULATION

You must complete this task on the following cluster/nodes: Cluster: immutable-cluster
Master node: master1
Worker node: worker1

You can switch the cluster/configuration context using the following command: [desk@cli] $ kubectl config use-context immutable-cluster

Context: It is best practice to design containers to be stateless and immutable.

Task:
Inspect Pods running in namespace prod and delete any Pod that is either not stateless or not immutable.

Use the following strict interpretation of stateless and immutable:

1. Pods being able to store data inside containers must be treated as not stateless.

Note: You don't have to worry whether data is actually stored inside containers or not already.

2. Pods being configured to be privileged in any way must be treated as potentially not stateless or not immutable.

Answer: See the explanation below

Explanation:

```
candidateScii:-? kubectl config use context KSRS00501 Switched to context "KSRS0Q501". candidateScli:~$ kubectl get pod —n
testing NAME          READY     STATUS          RESTARTS AGE
app           1/1       Running     0               6h31m
frontend 1/1          Running     0               6h32m
smtp          1/1       Running     0              61131m
candidateScli:~? kubectl get pod/app -n testing -o yaml - lastProbeTime: null lastTransitionTime: "2022-05-20T08 :«:35Z‘ status!
"True" type: PodScheduled containerstatuses: - containerlD:
docker://11143682c4C0984c9faf3dffle056d4b00a7eblde007fel834be0a84fal46el8 image: nginx:latest imagelD: docker-
pullable://nginx@sha256:2dl7cc4981bfIe22a87ef3b3dd20fbb72c386B738e3f3076 62eb40e2630d4320 laststate: {} name: app-
container ready: true restartCount: 0 started: true state: running: startedAt: "20:22-05-201’08:40:372" hostIP: 10.240.86.141 phase:
Running podIP: 10.10.1.3 podiPs: - ip: 10.10.1.3 qosclass: BestEffort startTime: "2022-05-20T08:40:35Z"
candidate@cli:-$ kubectl get pod/app -n testing -o yaml | grep -E 'privilegedIReadonlyFileSy stem' privileged: true
candidatefcli:~$ kubectl get pod/frontend -n testing -o yaml 1 grep -E 'privileged IReadOnlyF ileSystem’ privileged: false
```

```
candidate@cli:~$ kubectl get pod/smtp -n testing -o yaml I grep -E 'privileged|ReadOnlyFileS ystem' privileged: true
candidate@cii:~$ kubectl get pod -n testing -o yaml | grep -i Readonly readOnlyRebtFilesystem: false readOn y: true
                readOniyRoptFilesystem: true readonly: true :-?.-;3>. RootFilesystem: false readonly: true
candlldate@cli:'$ kubectl get pod/smtp -n testing -o yaml | grep -E 'privileged!readonlyRoolF ileSystem' privileged: true

candidate@cli:~$ kubectl get pod/app -n testing -o yaml i grep -E 'privileged!readOnlyRootFi lesystem' privileged: true
candidate@cli:~$ kubectl get pod/frontend -n testing -o yaml i grep -E 'privilegedIreadOnlyR ootFileSystem' privileged: false
candidate@cli:~? kubectl get pod/frontend -n testing -o yaml I grep -E 'privileged|readOnlyR ootFllesystem’ privileged: true
readOnlyRootFilesystem: false
candidate@cli:~? kubectl delete pod/app -n testing pod "app" deleted
candidate®cli:~$ kubectl get pod/smtp -n testing -o yaml | grep -E 'privileged,readOnlyRootF ilesystern' privileged: true
reactinlyRootFilesysrein: false
candidate@cii:~$ kubectl delete pod/smtp -n testing pod "smtp" deleted
```

Reference: https://kubernetes.io/docs/concepts/policy/pod-security-policy/https://cloud.google.com/architecture/best-practices-for-operating-containers

Question: 30

SIMULATION

You can switch the cluster/configuration context using the following command: [desk@cli] $ kubectl config use-context stage

Context:

A PodSecurityPolicy shall prevent the creation of privileged Pods in a specific namespace.

Task:

1. Create a new PodSecurityPolcy named deny-policy, which prevents the creation of privileged Pods.
2. Create a new ClusterRole name deny-access-role, which uses the newly created PodSecurityPolicy deny-policy.
3. Create a new ServiceAccount named psd-denial-sa in the existing namespace development.

Finally, create a new ClusterRoleBindind named restrict-access-bind, which binds the newly created ClusterRole deny-access-role to the newly created ServiceAccount psp-denial-sa

Answer: See the explanation below

Explanation:

www.atmicnetworks.com

Create psp to disallow privileged container

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole

metadata:

name: deny-access-role

rules:

- apiGroups: ['policy']

resources: ['podsecuritypolicies']

verbs: ['use']

resourceNames:

- "deny-policy"

k create sa psp-denial-sa -n development

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: restrict-access-bing

roleRef:

kind: ClusterRole

name: deny-access-role

apiGroup: rbac.authorization.k8s.io

subjects:

- kind: ServiceAccount

name: psp-denial-sa

namespace: development

Explanation

master1 $ vim psp.yaml

www.atmicnetworks.com

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
name: deny-policy
spec:
privileged: false # Don't allow privileged pods!
seLinux:
    rule: RunAsAny
supplementalGroups:
    rule: RunAsAny
runAsUser:
    rule: RunAsAny
fsGroup:
    rule: RunAsAny
volumes:
    '*'
-
```

```
master1 $ vim cr1.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
name: deny-access-role
rules:
-   apiGroups: ['policy']
resources: ['podsecuritypolicies']
verbs:    ['use']
```

www.atmicnetworks.com

resourceNames:

- “deny-policy”

master1 $ k create sa psp-denial-sa -n development

master1 $ vim cb1.yaml

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: restrict-access-bing

roleRef:

kind: ClusterRole

name: deny-access-role

apiGroup: rbac.authorization.k8s.io

subjects:

Authorize specific service accounts:

- kind: ServiceAccount

name: psp-denial-sa

namespace: development

master1 $ k apply -f psp.yaml
master1 $ k apply -f cr1.yaml
master1 $ k apply -f cb1.yaml

Reference: https://kubernetes.io/docs/concepts/policy/pod-security-policy/

Question: 31
SIMULATION

Context:
Cluster: prod
Master node: master1

Worker node: worker1

You can switch the cluster/configuration context using the following command:

[desk@cli] $ kubectl config use-context prod

Task:

Analyse and edit the given Dockerfile (based on the ubuntu:18:04 image) /home/cert_masters/Dockerfile fixing two instructions present in the file being prominent security/best-practice issues.

Analyse and edit the given manifest file /home/cert_masters/mydeployment.yaml fixing two fields present in the file being prominent security/best-practice issues.

www.atmicnetworks.com

www.atmicnetworks.com

Note: Don't add or remove configuration settings; only modify the existing configuration settings, so that two configuration settings each are no longer security/best-practice concerns.

Should you need an unprivileged user for any of the tasks, use user nobody with user id 65535

Answer: See the explanation below

Explanation:

1. For Dockerfile: Fix the image version & user name in Dockerfile
2. For mydeployment.yaml : Fix security contexts

Explanation

[desk@cli] $ vim /home/cert_masters/Dockerfile

FROM ubuntu:latest # Remove this

FROM ubuntu:18.04 # Add this

USER root # Remove this

USER nobody # Add this

RUN apt get install -y lsof=4.72 wget=1.17.1 nginx=4.2

ENV ENVIRONMENT=testing

USER root # Remove this

Add this

USER nobody

CMD ["nginx -d"]

```
FROM ubuntu:latest    #  Remove this
FROM ubuntu:18.04     #  Add this
USER root             #  Remove this
USER nobody           #  Add this
RUN apt get install -y lsof=4.72 wget=1.17.1 nginx=4.2
ENV ENVIRONMENT=testing
USER root            # Remove this
USER nobody          # Add this
CMD ["nginx -d"]
```

[desk@cli] $ vim /home/cert_masters/mydeployment.yaml

www.atmicnetworks.com

```
apiVersion: apps/v1
kind: Deployment
metadata:
creationTimestamp: null
labels:
app: kafka
name: kafka
spec:
replicas: 1
selector:
matchLabels:
app: kafka
strategy: {}
template:
metadata:
creationTimestamp: null
labels:
app: kafka
spec:
containers:
- image: bitnami/kafka
name: kafka
volumeMounts:
- name: kafka-vol
mountPath: /var/lib/kafka
securityContext:
```

{"capabilities":{"add":["NET_ADMIN"],"drop":["all"]},"privileged":
True,"readOnlyRootFilesystem": False, "runAsUser": 65535} # Delete This

{"capabilities":{"add":["NET_ADMIN"],"drop":["all"]},"privileged":
False,"readOnlyRootFilesystem": True, "runAsUser": 65535} # Add This

resources: {}

volumes:

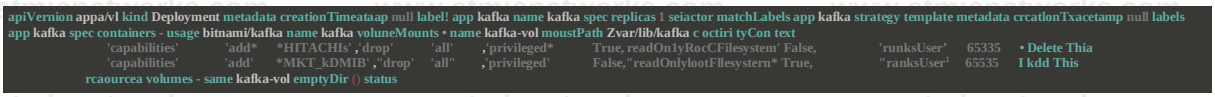
- name: kafka-vol

emptyDir: {}

status: {}

Pictorial View:

[desk@cli] $ vim /home/cert_masters/mydeployment.yaml



Reference: https://kubernetes.io/docs/concepts/policy/pod-security-policy/

Question: 32

SIMULATION

You can switch the cluster/configuration context using the following command: [desk@cli] $ kubectl config use-context test-account

Task: Enable audit logs in the cluster.

To do so, enable the log backend, and ensure that:

1. logs are stored at /var/log/Kubernetes/logs.txt
2. log files are retained for 5 days
3. at maximum, a number of 10 old audit log files are retained

A basic policy is provided at /etc/Kubernetes/logpolicy/audit-policy.yaml. It only specifies what not to log.

Note: The base policy is located on the cluster's master node.

Edit and extend the basic policy to log:

1. Nodes changes at RequestResponse level
2. The request body of persistentvolumes changes in the namespace frontend
3. ConfigMap and Secret changes in all namespaces at the Metadata level

Also, add a catch-all rule to log all other requests at the Metadata level

Note: Don't forget to apply the modified policy.

Answer: See the explanation below

Explanation:

$ vim /etc/kubernetes/log-policy/audit-policy.yaml

```
- level: RequestResponse
  userGroups: ["system:nodes"]
- level: Request
  resources:
  - group: "" # core API group
    resources: ["persistentvolumes"]
    namespaces: ["frontend"]
- level: Metadata
  resources:
  - group: ""
    resources: ["configmaps", "secrets"]
- level: Metadata
```

$ vim /etc/kubernetes/manifests/kube-apiserver.yaml

Add these

```
- --audit-policy-file=/etc/kubernetes/log-policy/audit-policy.yaml
- --audit-log-path=/var/log/kubernetes/logs.txt
- --audit-log-maxage=5
```

www.atmicnetworks.com

- --audit-log-maxbackup=10

Explanation

[desk@cli] $ ssh master1
[master1@cli] $ vim /etc/kubernetes/log-policy/audit-policy.yaml

apiVersion: audit.k8s.io/v1 # This is required.

kind: Policy

- Don't generate audit events for all requests in RequestReceived stage.

omitStages:

- "RequestReceived"

rules:

- Don't log watch requests by the "system:kube-proxy" on endpoints or services

- level: None

users: ["system:kube-proxy"]

verbs: ["watch"]

resources:

- group: "" # core API group

resources: ["endpoints", "services"]

- Don't log authenticated requests to certain non-resource URL paths.

- level: None

www.atmicnetworks.com

```
    userGroups: ["system:authenticated"]
  nonResourceURLs:
  - "/api*" # Wildcard matching.
  - "/version"

  # Add your changes below

  - level: RequestResponse
    userGroups: ["system:nodes"]                    # Block for nodes
  -level: Request
  resources:
  - group: "" # core API group                      # Block for persistentvolumes
    resources: ["persistentvolumes"]                # Block for persistentvolumes of frontend ns
    namespaces: ["frontend"]

  -level: Metadata
  resources:
  - group: "" # core API group                      # Block for configmaps & secrets
    resources: ["configmaps", "secrets"]
  - level: Metadata                                 # Block for everything else

[master1@cli] $ vim /etc/kubernetes/manifests/kube-apiserver.yaml
apiVersion: v1
```

kind: Pod

metadata:

annotations:

kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.0.0.5:6443

labels:

component: kube-apiserver

tier: control-plane

name: kube-apiserver

namespace: kube-system

spec:

containers:

- command:
- kube-apiserver
- --advertise-address=10.0.0.5
- --allow-privileged=true
- --authorization-mode=Node,RBAC
- --audit-policy-file=/etc/kubernetes/log-policy/audit-policy.yaml #Add this
- --audit-log-path=/var/log/kubernetes/logs.txt #Add this
- --audit-log-maxage=5 #Add this
- --audit-log-maxbackup=10 #Add this

output truncated

Note: log volume & policy volume is already mounted in vim /etc/kubernetes/manifests/kube-apiserver.yaml so no need to mount it.

Reference: https://kubernetes.io/docs/tasks/debug-application-cluster/audit/

Question: 33
SIMULATION

You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSSHOO	kssh00401	kssh00401
401	-master	-workerl

You can switch the cluster/configuration context using the following command:

[candidate@cli] $ kubec tl config use-context KS SH00401

Context
AppArmor is enabled on the cluster's worker node. An AppArmor profile is prepared, but not enforced yet.

You may use your browser to open **one additional tab** to access the AppArmor documentation.

Task

On the cluster's worker node, enforce the prepared AppArmor profile located at /etc/apparmor.d/nginx_apparmor.

Edit the prepared manifest file located at /home/candidate/KSSH00401/nginx-pod.yaml to apply the AppArmor profile.

Finally, apply the manifest file and create the Pod specified in it.

Answer: See the explanation below

Explanation:

```
candidate@cli:~5 kubectl config use-context KSSH00401 Switched to context "KSSH00401".
candidate@cli;~$ ssh kssh00401-workerl
Warning: Permanently added '10.240.86.172' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

root@kssh00401-workerl:~# head /etc/apparmor.d/nginx_apparmor Sinclude <tunables/global>

profile nginx profile-2 flags=(attach_disconnected,mediate_deleted) {
  #include <abstractipns/base> network inet tcp, network inet udp, network inet icmp,
```

```
    deny network raw, root@kssh00401-workerl:~i apparmor parser -q /etc/appannor.d/nginx apparmor
root@kssh004UL workerl:-# exit logout Connection to 10.240.86.172 closed.

candidate@cli:~$ cat KSSH00401/nginx-pod.yaml

apiVersion: vl kind: Pod metadata: name: nginx-pod spec: containers: name: nginx-pod image: nginx:1.19.0
ports: - containerPort: 80
candidate@cli:~$ vim KSSHO04O1/nginx-pod.yaml]
```

```
                vl Pod W6tdddt8I nginx-pod sunotations:
                                                    localhost/nginx-pi spec:
  containers:
  -           nginx-pod
                nginx:1.19.0 ports: - eontainerPort:
```

```
candidate@cli:'$ vim KSSH00401/nginx pod.yaml
candidategcli:~? kubectl create -f KSSH00401/nginx pod.yaml pod/nginx-pod created
candidate@cli:~$ cat KSSH00401/nginx-pod.yaml

apiVersion: vl kind: Pod metadata:
    same: nginx-pod
    annotations:
      container, appannor.security.beta.kubernetes.io/ng i nx-pod: localhost/nginx-profile-2 spec: containers: - name: nginx-pod
      image: nginx:1.19.0 ports: - containerPort: 80
```

www.atmicnetworks.com

Reference: https://kubernetes.io/docs/tutorials/clusters/apparmor/

Question: 34

SIMULATION

You can switch the cluster/configuration context using the following command: [desk@cli] $ kubectl config use-context qa

Context:

A pod fails to run because of an incorrectly specified ServiceAccount

Task:

Create a new service account named backend-qa in an existing namespace qa, which must not have access to any secret.

Edit the frontend pod yaml to use backend-qa service account

Note: You can find the frontend pod yaml at /home/cert_masters/frontend-pod.yaml

Answer: See the explanation below

Explanation:

[desk@cli] $ k create sa backend-qa -n qa sa/backend-qa created

[desk@cli] $ k get role,rolebinding -n qa

No resources found in qa namespace.

[desk@cli] $ k create role backend -n qa --resource pods,namespaces,configmaps --verb list

No access to secret

[desk@cli] $ k create rolebinding backend -n qa --role backend --serviceaccount qa:backend-qa

[desk@cli] $ vim /home/cert_masters/frontend-pod.yaml

apiVersion: v1

kind: Pod

metadata:

name: frontend

www.atmicnetworks.com

spec:

serviceAccountName: backend-qa # Add this

image: nginx

name: frontend

[desk@cli] $ k apply -f /home/cert_masters/frontend-pod.yaml

pod created

[desk@cli] $ k create sa backend-qa -n qa

serviceaccount/backend-qa created

[desk@cli] $ k get role,rolebinding -n qa

No resources found in qa namespace.

[desk@cli] $ k create role backend -n qa --resource pods,namespaces,configmaps --verb list

role.rbac.authorization.k8s.io/backend created

[desk@cli] $ k create rolebinding backend -n qa --role backend --serviceaccount qa:backend-qa

rolebinding.rbac.authorization.k8s.io/backend created

[desk@cli] $ vim /home/cert_masters/frontend-pod.yaml

apiVersion: v1

kind: Pod

metadata:

name: frontend

spec:

serviceAccountName: backend-qa # Add this

image: nginx

name: frontend

[desk@cli] $ k apply -f /home/cert_masters/frontend-pod.yaml

pod/frontend created

https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

www.atmicnetworks.com

Question: 35

SIMULATION

You must complete this task on the following cluster/nodes:

Cluster: trace

Master node: master

Worker node: worker1

You can switch the cluster/configuration context using the following command: [desk@cli] $ kubectl config use-context trace

Given: You may use Sysdig or Falco documentation.

Task:

Use detection tools to detect anomalies like processes spawning and executing something weird frequently in the single container belonging to Pod tomcat.

Two tools are available to use:

1. falco
2. sysdig

Tools are pre-installed on the worker1 node only.

Analyse the container's behaviour for at least 40 seconds, using filters that detect newly spawning and executing processes.

Store an incident file at /home/cert_masters/report, in the following format:

[timestamp],[uid],[processName]

Note: Make sure to store incident file on the cluster's worker node, don't move it to master node.

Answer: See the explanation below

Explanation:

$vim /etc/falco/falco_rules.local.yaml

- rule: Container Drift Detected (open+create)

desc: New executable created in a container due to open+create

condition: >

evt.type in (open,openat,creat) and

evt.is_open_exec=true and

www.atmicnetworks.com

container and

not runc_writing_exec_fifo and

not runc_writing_var_lib_docker and

not user_known_container_drift_activities and

evt.rawres>=0

output: >

%evt.time,%user.uid,%proc.name # Add this/Refer falco documentation

priority: ERROR

$kill -1 <PID of falco>

Explanation

[desk@cli] $ ssh node01

[node01@cli] $ vim /etc/falco/falco_rules.yaml

search for Container Drift Detected & paste in falco_rules.local.yaml

[node01@cli] $ vim /etc/falco/falco_rules.local.yaml

- rule: Container Drift Detected (open+create)

desc: New executable created in a container due to open+create

condition: >

evt.type in (open,openat,creat) and

evt.is_open_exec=true and

container and

not runc_writing_exec_fifo and

not runc_writing_var_lib_docker and

not user_known_container_drift_activities and

evt.rawres>=0

output: >

%evt.time,%user.uid,%proc.name # Add this/Refer falco documentation

priority: ERROR

[node01@cli] $ vim /etc/falco/falco.yaml

```
file_output:
    enabled: true keep_alive: false filename: /home/cert_masters/report
```

send HUP signal to falco process to re-read the configuration

```
root^node0l:/etc/falco# pa -ef | grep falco
root     10127       1 1 17:13 ?        00:00:35 /usr/bin/falco     –pidfile«/var/run/falco.pid    -c /etc/falco/falco.yaml
root     30938   20175 0 17:55 pta/l    00:00:00 grep falco
root8node01:/etc/falco# kill -1 10127
```

Reference:
https://falco.org/docs/alerts/
https://falco.org/docs/rules/supported-fields/

Question: 36

SIMULATION

Cluster: dev

Master node: master1
Worker node: worker1

You can switch the cluster/configuration context using the following command:

[desk@cli] $ kubectl config use-context dev

Task:

Retrieve the content of the existing secret named adam in the safe namespace.

Store the username field in a file names /home/cert-masters/username.txt, and the password field in a file named /home/cert-masters/password.txt.

1. You must create both files; they don't exist yet.
2. Do not use/modify the created files in the following steps, create new temporary files if needed.

Create a new secret names newsecret in the safe namespace, with the following content: Username: dbadmin

Password: moresecurepas

Finally, create a new Pod that has access to the secret newsecret via a volume:

Namespace: safe

www.atmicnetworks.com

Pod name: mysecret-pod

Container name: db-container

Image: redis

Volume name: secret-vol

Mount path: /etc/mysecret

Answer: See the explanation below

Explanation:

```
candidateScli:-5 kubectl config use-context KSMV00201
Switched to context "KSMV00201".
candidate@cli:~? kubectl get secret n monitoring
NAME                  TYPE                                  DATA    AGE
dbl-test              Opaque                                2       6h23m
default-token-cqqf6   kubernetes.io/service-account- token  3       6h23m
candidategcll:~$ kubectl get secret/dbl-test -n monitoring
NAME       TYPE       DATA AGE
dbl-test   Opaque     2    6h23m
candidate@cli:~5 kubectl get secret/dbl-test -n monitoring o yaml
apiVersion: vl da ta:
    password: QVU3dHblbXFOTHZt
    username: cHJvZHVjdGlvbiOx
kind: Secret
metadata:
    creationTimestamp: "2022-05-20T08:37:33Z"
    name: dbl-test
    namespace: monitoring
    resourceversion: "2586"
    Uid: 659bd4ac-e0ba-4d9f-b411-816f2aedf7e6
type: Opaque
candidate@cli:-$ echo "cHJvZHVjdGlvbiOx" I base64 -d
production-1 candidate@cli:~$
candida te@c1i:~$
candidate@cli:~9 echo "cHJvZHVjdGlvbiOx" | base64 -d > /home/candidate/username.txt
candidate@cli:~S cat /home/candidate/username.txt
productipn-lcandidateScii:~$
candidateScli:~$
candida Le@cli:~$
candidate^!i:~S echo "QVU3dHhlbXEOTHZt" I base64 -d
AU7txumqNLvmcandidate0cli:~$ echo "QTO3dHhlbXFOTHZt" | base64 -d > /home/candidate/password. txt candidate@cli:-$
cat /home/candidate/password.txt
AU7txumqNLvmcandidate@cli:~$
candidateScii:~$
candidate@cli:~S
candidate0cll:~$ kubectl create secret generic test-workflow —from-literal=username=dev-dat
abase —from-literal=password=aV?HR7nU3JLx -n monitoring
secret/test-workflow created
candidate@cli:~?
candidate@cli:~$
candidate@cli:-$ kubectl -n monitoring run test-secret-pod —image=httpd --dry-run=client - o yaml > test-secret-pod.yaml
candidate@cli:~$ vim test-secret-pod.yaml,
```

: vl

Pod

■H u 3. LS •

test-secret-pod
test-secret-pod
monitoring

V 4
olumes:

dev-volume

test-workflow

ontainers:

httpd
dev-container

dev vo 11line
/et 7"*i-d-rJ. id 1 - ClusterFirst
A1 wa y s

candidate@cli:~$ kubectl -n monitoring run test-secret-pod - image-httpd ■ dry run-client - o yaml > test-secret-pod,yaml
candidate@cli:~$ vim test-secret-pod.yaml
candidate®cli:~$ cat test-secret-pod.yaml!

```
   labels:
     run: test-secret-pod name: test-secret-pod namespace:
monitoring spec:
    volumes: - name: dev-volume secret: secretName: test-
    workflow containers: - image: httpd name: dev-container
    resources: {} volumeMounts:
        -   name:   dev-volume   mountPath:   /etc/credentials
dnsPolicy: ClusterFirst restartPol icy: Always status: {}
candidate@cli:~$   kubectl   create   -f   test-secret-pod.yaml
pod/test-secret-pod created candidate@cli:*$ kubectl get pods
-n monitoring NAME       READY     STATUS     RESTARTS AGE
test-secret-pod   1/1     Running   0          9s
```

www.atmicnetworks.com

```
candidate@cli:-$ b
```

Question: 37
SIMULATION

Cluster: scanner
Master node: controlplane
Worker node: worker1

You can switch the cluster/configuration context using the following command:

[desk@cli] $ kubectl config use-context scanner

Given:
You may use Trivy's documentation.

Task:
Use the Trivy open-source container scanner to detect images with severe vulnerabilities used by Pods in the namespace nato.

Look for images with High or Critical severity vulnerabilities and delete the Pods that use those images.

Trivy is pre-installed on the cluster's master node. Use cluster's master node to use Trivy.

Answer: See the explanation below

Explanation:

```
candidate8cli:~$ kubectl config use-context KSSC00401
Switched to context "KSSC00401".
candidate@cli:~$ ash kssc00401-master
Warning: Permanently added '10.240.86.231' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
```

```
root9kssc00401-master:~i kubectl get pods -n naboo
NAME            READY   STATUS    RESTARTS   AGE
c-3po           1/1     Running   0          6h48m
chewbacca       1/1     Running   0          6h48m
jawas           1/1     Running   0          6h48m
qui-gon-jinn    1/1     Running   0          6h48m
root@kssc00401 master:'i kubectl get pods -n naboo -o name
pod/c-3po
pod/chewbacca
pod/jawas
pod/qui-gon-j inn
root@kssc00401-master:'l for i in $(kubectl get pods -n naboo -o name)
> do
> kubectl get ${i}      -o yaml |      grep -i image
> done Error from
server                  (NotFound)  : pods "c-3po" not found
Error from server       (NotFound)  : pods "chewbacca" not found
Error from server       (NotFound)  : pods "jawas" not found
Error from server       (NotFound)  : pods "qui-gon-jinn" not found
root8kssc00401-master:~i for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo
get 5(i) o yaml I grep -i image ; done
      imaga: centos:centos7.9.2009
      imagePullPolicy: Never
      image: centos:centos7.9.2009
      naagalD: docker pullable://centosesha256:c73f515d06b0fa07bbl8dB202035e739a494ce760aa7312 9f60f4bf2bd22b407
      image: photon:3.0
        -..•.,pullpolicy: Never
      imago: photon:3.0
      imagalD: docker-pullable://photonesha256:c48d61f0f3adl9215b75e2087cfbe95d7321abb454e4295 a0e6c38f563e ce622
      image: alplne:3.7
      imaqoPullPolicy: Never
      image: alpine:3.7
        - -ID: docker-pullable://alpineesha256:8421d9a84432575381bfabd248fleb56f3aa21d9d7cd251 1583c68c9b7511dl0
      image: amazonlinux:2
      imagePullPollcy: Never
      image: amazonlinux:2
      imagelD: docker-pullable://amazonlinux?sha256:246ef631c75ea83005889621119fd5cc9cbb5500el 93707
c38b6c060d597a146
root?kssc00401 master:'! trivy image ccntos:centos7.9.2009
2022 05-20T15:39:51.7332                    Need to update DB
2022-05-20T15:39:51.733Z                    Downloading DB...
27.97 MiB / 27.97 MiB [ ----------------------------------------------------------------1 100.00% 27.43 MiB p/s Is

--------------------------+ ----------------------------------------------------------------------------------------+ root@kssc00401-
master;-4 for i in $ (kubectl get pods -n naboo -o name); do kubectl -n naboo get ${1} -o yaml | grep -i image ; done
      image: centos:centos?.9.2009
      imagepulipoliqy; Never
      -       : centos:centos?.9.2009
      imagelD: docker-pullable:Z/centos@sha256:c73f515d06b0fa07bbl8d8202035e739a494ce760aa7312 9f60f4bf2bd22b407
      image: photon:3.0
            >?::] IPol i cy: Never
      image: photon:3.0
      imagmID! docker-pullable://photonGsha256:c4Bd61f0f3adl9215b75e2087cfbe95d7321abb454e4295 aOe 6c38 f563ece
622
      image: al pine:3.7
      imagePuHPolicy: Never
      image: alpine:3.7
      imagelD; docker-pullable://alpine0sha256:8421d9a844325753Blbfabd248fLeb56f3aa21d9d7cd251 1583c68c9b7511dl0
      image: amazonlinux:2
      imagePuLlPgliiey: Never
```

```
imager amazonlinux:2

. ID: docker-puliable://amazonlinux?sha256:246ef631c?5ea83005889621119fd5cc9cbb5S00el 93707c38b
6c060d597 a 146 root@kssc00401-master:-4 trivy image photon:3.0 2022-05-20115:40:18.0032 " Detected OS:
photon
2022-05-20T15:40:18.0032 Detecting Photon Linux vulnerabilities...

2022-05-20T15:40:18.005Z Number of language-specific files: 0

photon:3.0 (photon 3.0)

Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)
```

```
root@kssc00401~master:~4 kubectl get pods -n naboo -o name
pod/c-3po
pod/chewbacca
pod/jawas
pod/qui-gon-j inn
root@kssc0Q401-master:-4 kubectl -n naboo pod/c-3po -o yaml | grep image
Error: flags cannot be placed before plugin name: -n
root@kssc00401-master:-I kubectl -n naboo get pod/c-3po -o yaml | grep image
    image: centos:centos?.9.2009
    imaqi pull Policy: Never
    image: centos:centos?.9.2009
    imag«D: docker-pullable://centos0sha256:c73f515d06b0ta07bbl8dB202035e739a494ce?60aa7312 9f6.0f4bf2bd22b407
root@kssc00401-master:-4 kubectl -n naboo delete pod/c-3pO
pod "c-3po" deleted
root@kssc00401-master;~4 kubectl -n naboo delete pod/jawas pod "jawas" deleted
pod "jawas" deleted
root@ksscQ0401-master:^1 history
    1 kubectl get pods -n taboo
    2 kubectl get pods -n taboo -o name
    3 for i in $(kubectl get pods -n naboo -o name); do kubectl get $(i)-o yaml | grep -
image ; done
    4 for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo get $ (i} -o yaml
grep -i image ; done
    5 trivy image centos:centos?.9.2009
    6 for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo get $( 1] -o yaml
grep -i image ; done
    7 trivy image photon:3.0
    3 for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo get $(i} -o yaml
-i image ; done
grep 9 trivy image alpine:3.7
    10 for i in $(kubectl get pods -n naboo -o name); do kubectl -n naboo get $(i) -o yaml
grep -i image ; done
    11 trivy image amasotlinux:2
    12 kubectl get pods -n taboo -o name
    13 kubectl -n naboo pod/c~3po -o yaml | grep image
    14 kubectl n naboo get pod/c 3po o yarn! | grep image
    15 kubectl -n naboo delete pod/c-3po
    16 kubectl -n taboo delete pod/jawas
    17 history
root@ksscOQ4Ul-master:~# Q
```

Reference: https://github.com/aquasecurity/trivy

www.atmicnetworks.com

Question: 38

SIMULATION

You can switch the cluster/configuration context using the following command: [desk@cli] $ kubectl config use-context dev

A default-deny NetworkPolicy avoid to accidentally expose a Pod in a namespace that doesn't have any other NetworkPolicy defined.

Task: Create a new default-deny NetworkPolicy named deny-network in the namespace test for all traffic of type Ingress + Egress

The new NetworkPolicy must deny all Ingress + Egress traffic in the namespace test.

Apply the newly created default-deny NetworkPolicy to all Pods running in namespace test.

You can find a skeleton manifests file at /home/cert_masters/network-policy.yaml

Answer: See the explanation below

Explanation:

master1 $ k get pods -n test --show-labels

NAME READY STATUS RESTARTS AGE LABELS

test-pod 1/1 Running 0 34s role=test,run=test-pod

testing 1/1 Running 0 17d run=testing

$ vim netpol.yaml

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: deny-network

namespace: test

spec:

podSelector: {}

www.atmicnetworks.com

policyTypes:

- Ingress

- Egress

master1 $ k apply -f netpol.yaml

Explanation

controlplane $ k get pods -n test --show-labels

NAME READY STATUS RESTARTS AGE LABELS

test-pod 1/1 Running 0 34s role=test,run=test-pod

testing 1/1 Running 0 17d run=testing

master1 $ vim netpol1.yaml

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: deny-network

namespace: test

spec:

podSelector: {}

policyTypes:

- Ingress

- Egress

master1 $ k apply -f netpol1.yaml

Reference:

https://kubernetes.io/docs/concepts/services-networking/network-policies/

Explanation

www.atmicnetworks.com

controlplane $ k get pods -n test --show-labels

NAME READY STATUS RESTARTS AGE LABELS

test-pod 1/1 Running 0 34s role=test,run=test-pod

testing 1/1 Running 0 17d run=testing

master1 $ vim netpol1.yaml

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: deny-network

namespace: test

spec:

podSelector: {}

policyTypes:

- Ingress
- Egress

master1 $ k apply -f netpol1.yaml

Reference:

https://kubernetes.io/docs/concepts/services-networking/network-policies/

Question: 39

SIMULATION

Context:

Cluster: gvisor

Master node: master1

Worker node: worker1

You can switch the cluster/configuration context using the following command:

[desk@cli] $ kubectl config use-context gvisor

www.atmicnetworks.com

Context: This cluster has been prepared to support runtime handler, runsc as well as traditional one.

Task:

Create a RuntimeClass named not-trusted using the prepared runtime handler names runsc.

Update all Pods in the namespace server to run on newruntime.

Answer: See the explanation below

Explanation:

1. Create runtime class by the name of not-trusted using runsc handler

```
apiversicn: noae.K8s.io/vl kind: Runtimeclass metadata:
  name: not-trusted handler: runsc
```

2. Find all the pods/deployment and edit runtimeClassName parameter to not-trusted under spec

[desk@cli] $ k edit `deploy nginx`

```
spec: runtimeclassware: not-trusted. - Add this
```

Explanation

[desk@cli] $vim runtime.yaml

apiVersion: node.k8s.io/v1

kind: RuntimeClass

metadata:

name: not-trusted

handler: runsc

[desk@cli] $ k apply -f runtime.yaml

[desk@cli] $ k get pods

NAME	READY	STATUS	RESTARTS	AGE
nginx-6798fc88e8-chp6r	1/1	Running	0	11m
nginx-6798fc88e8-fs53n	1/1	Running	0	11m
nginx-6798fc88e8-ndved	1/1	Running	0	11m

[desk@cli] $ k get deploy

NAME	READY	UP-TO-DATE	AVAILABLE	AGE

www.atmicnetworks.com

nginx 3/3 11 3 5m

[desk@cli] $ k edit deploy nginx

```
apiVersion apps/vl kind Deployment metadata labels app nginx
name nginx spec replicas 3 selector: matchLabels app nginx
strategy {} template: metadata: labels app nginx spec
runtimeClassName not-trusted # Add this containers - image
nginx name nginx resources: {} status
```

Reference: https://kubernetes.io/docs/concepts/containers/runtime-class/

Question: 40
SIMULATION

You can switch the cluster/configuration context using the following command: [desk@cli] $ kubectl config use-context prod-account

Context:

A Role bound to a Pod's ServiceAccount grants overly permissive permissions. Complete the following tasks to reduce the set of permissions.

Task:

Given an existing Pod named web-pod running in the namespace database.

1. Edit the existing Role bound to the Pod's ServiceAccount test-sa to only allow performing get operations, only on resources of type Pods.

2. Create a new Role named test-role-2 in the namespace database, which only allows performing update operations, only on resources of type statuefulsets.

3. Create a new RoleBinding named test-role-2-bind binding the newly created Role to the Pod's ServiceAccount.

Note: Don't delete the existing RoleBinding.

Answer: See the explanation below

Explanation:

www.atmicnetworks.com

```
candidate@cli:~$ kubectl config use-context KSCH00201 Switched to context "KSCH00201". candidategcii:~$ kubectl get pods -
n security NAME          READY        STATUS        RESTARTS          AGE
web-pod      1/1          Running       0                 6h9m
candidate@cli:~$ kubectl get deployments.apps -n security No resources found in security namespace.
candidateScli:~$ kubectl describe rolebindings.rbac.authorization.kBs.io -n security Name:      dev-role
labels:                <none>
Anhota t ions: <none> Role:
     Kind: Role
     Name: dev-role
Subjects:
     Kind                      Name Namespace

     ServiceAccount sa-dev-1
candidate?cli:~$ kubectl describe role dev-role n security
Name:                dev-role
Labels:              <none>
Annotations: <none>
PolicyRule:
     Resources Non-Resource URLs Resource Names Verbs

     *                [J                    []                  [*]
candidate@cli:~$ kubectl edit role/dev-role -n security |
```

b4c9ddd6-2729-43bd-8fbd-b2d227f4c4cd

services

watch

```
candidate9cli:~$ kubectl describe role dev-role -n security
Name:          dev-role
Labels:        <none>
Annotations: <none>
PolicyRule:
    Resources Non-Resource URLs Resource Names Verbs

    *             [1                [1              [*l
candidate^cli:-$ kubectl edit role/dev-role -n security role.rbac.authorization.kBs.io/dev-role edited candidateScii:-? kubectl
describe role dev-role -n security Name:.  dev-role
Labels:              <none>
Annotations: <none> PolicyRule: Resources Non-Resource URLs Resource Names Verbs services []      []          [watch]
candidateScli:~5 kubectl get pods -h security NAME              READY   STATUS  RESTARTS          AGE
web-pod     1/1          Running       0                 6hl2m
candidate@cli:-$ kubectl get pods/web-pod n Security -o yaml ( grep serviceAccount serviceAccount: sa-dev-1 servi ' 'Name: sa-
dev-1 - serviceAccountTOkeni
candidate@cli:~S kubectl create role role-2 --verb-update —resource-namespaces -n security role.rbac.authorization.kBs.io/role-2
created candidate@cli:'$ kubectl create rolebinding role-2-binding —role —role       —role=
candidate@cli:~5 kubectl create rolebinding role-2-binding - role=role-2 --serviceaccount=se curity:sa-dev-1 -n security
rolebinding.rbac.authorization. kBs.io/role-2-binding created candidatc@cli:~$ Q
```

www.atmicnetworks.com

Question: 41

SIMULATION

You can switch the cluster/configuration context using the following command: [desk@cli] $ kubectl config use-context dev

Context:

A CIS Benchmark tool was run against the kubeadm created cluster and found multiple issues that must be addressed.

Task:

Fix all issues via configuration and restart the affected components to ensure the new settings take effect.

Fix all of the following violations that were found against the API server:

1.2.7 authorization-mode argument is not set to AlwaysAllow FAIL

1.2.8 authorization-mode argument includes Node FAIL

1.2.7 authorization-mode argument includes RBAC FAIL

Fix all of the following violations that were found against the Kubelet:

4.2.1 Ensure that the anonymous-auth argument is set to false FAIL

4.2.2 authorization-mode argument is not set to AlwaysAllow FAIL (Use Webhook autumn/authz where possible)

Fix all of the following violations that were found against etcd:

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

Answer: See the explanation below

Explanation:

worker1 $ vim /var/lib/kubelet/config.yaml

anonymous:

enabled: true #Delete this

enabled: false #Replace by this

authorization:

mode: AlwaysAllow #Delete this

mode: Webhook #Replace by this

worker1 $ systemctl restart kubelet. # To reload kubelet config

ssh to master1

master1 $ vim /etc/kubernetes/manifests/kube-apiserver.yaml

- -- authorization-mode=Node,RBAC

master1 $ vim /etc/kubernetes/manifests/etcd.yaml - --client-cert-auth=true

Explanation

ssh to worker1

worker1 $ vim /var/lib/kubelet/config.yaml

apiVersion: kubelet.config.k8s.io/v1beta1

www.atmicnetworks.com

```
authentication:
anonymous:
enabled: true                    #Delete this
 enabled: false                  #Replace by this
webhook:
cacheTTL: 0s
enabled: true
x509:
clientCAFile: /etc/kubernetes/pki/ca.crt
authorization:
mode: AlwaysAllow                #Delete this
mode: Webhook                    #Replace by this
webhook:
cacheAuthorizedTTL: 0s
cacheUnauthorizedTTL: 0s
cgroupDriver: systemd
clusterDNS:
- 10.96.0.10
clusterDomain: cluster.local
cpuManagerReconcilePeriod: 0s
evictionPressureTransitionPeriod: 0s
fileCheckFrequency: 0s
healthzBindAddress: 127.0.0.1
healthzPort: 10248
httpCheckFrequency: 0s
imageMinimumGCAge: 0s
```

kind: KubeletConfiguration

logging: {}

nodeStatusReportFrequency: 0s

nodeStatusUpdateFrequency: 0s

resolvConf: /run/systemd/resolve/resolv.conf

rotateCertificates: true

runtimeRequestTimeout: 0s

staticPodPath: /etc/kubernetes/manifests

streamingConnectionIdleTimeout: 0s

syncFrequency: 0s

volumeStatsAggPeriod: 0s

worker1 $ systemctl restart kubelet. # To reload kubelet config

ssh to master1

master1 $ vim /etc/kubernetes/manifests/kube-apiserver.yaml

```
apiversion vl kind Pod metadata annotations kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint 172.17.0.22:6443
labels component kube-apiserver tier control-plane name kube-apiserver namespace kube-system spec containers - command - kube-
apiserver - —advertise-address=172.17.0.22 - —allow-privileged= ue 5   - —authorization-mode"AlwaysAllow # Delete This
      - —authorization-mode=Node,RBAC          * Replace by this line
      - —client-ca-file=/etc/kubernetes/pki/ca.crt - —enable-admissior.-plugins=NodeRestriction - —enable-bootstrap-token-
auth=true - —etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt - —etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt - —
etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key - —etcd-servers=https://127.0.0.1:2379 - —insecure-port=0
```

master1 $ vim /etc/kubernetes/manifests/etcd.yaml

```
apiVersion vl kind Pod metadata annotations
    kubeadm.kubernetes.io/etcd.advertise-client-urls https://172.17.0.29:2379
  creationTimestamp null labels
    component etcd tier control-plane name etcd namespace kube-system spec containers
- command - etcd
    - —advertise-client-urls=https: //172.17.0.29:2379
    - —cert-file=/etc/kubernetes/pki/etcd/server.crt - —client-cert-auth=true #Add
    this line - —data—dir=/var/lib/etcd
    - —initial-advertise-peer-urls=https://172.17.0.29:2380 - —initial-
    cluster=controlplane=https://172.17.0.29:2380 - —key-
    file=/etc/kubernetes/pki/etcd/server.key
     --- listen-c1ient-urls=https://127.0.0.1:2379,https://172.17.0.29:2379
    - —listen-metrics-urls=http://127•0.0.1:2381 - —listen—peer—
    urls=https://172.17.0.29:2380 - —name=controlplane - —peer-cert-
    file=/etc/kubernetes/pki/etcd/peer.crt - —peer-client-cert-auth=true
    - —peer-key-file=/etc/kubernetes/pki/etcd/peer.key
    - —peer-trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt - —snapshot-count=10000
    - —trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt image k8s.gcr.io/etcd:3.4.9-l
    imagePullPolicy IfNotPresent
```

Reference:

kubelet parameters: https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet/ kubeapi parameters: https://kubernetes.io/docs/reference/command-line-tools-reference/kube- apiserver/

etcd parameters: https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/

Question: 42

SIMULATION

You must complete this task on the following cluster/nodes:

Master node	Worker node	Cluster
KSMVOO ksmv0030	ksmv0030	
301	1-master	1-worker1

You can switch the cluster/configuration context using the following command:

[candidatefdcli] $ kubec tl config use-context

www.atmicnetworks.com

KS MV00301

Context
This cluster uses containerd as CRI runtime.

Containerd's default runtime handler is runc. Containerd has been prepared to support an additional runtime handler, runsc (gVisor).

Task

Create a RuntimeClass named sandboxed using the prepared runtime handler named runsc.

Update all Pods in the namespace server to run on gVisor.

You can find a skeleton manifest file at /home/candidate/KSMV00301/runtime-class.yaml

Answer: See the explanation below

Explanation:

Explanation:

```
candidate@cli:~$ kubectl config use-context KSMV00301 Switched to context
"KSMV00301".
candidate@cli:~$ cat /home/candidate/KSMV00301/runtime-class.yaml

apiVersion: node.k8s.io/vl
kind: Runtimeclass
metadata:
   name: "" handler: ""
candidate@cli:~$ vim /home/candidate/KSMV00301/runtime-class.yaml|
```


```
candidate@cli:~$ kubectl config use-context KSMV00301
Switched to context "KSMV00301".
candidate@cli:~$ cat /home/candidate/KSMV00301/runtime-class.yaml
```

www.atmicnetworks.com

```
apiVersion: node.k8s.io/vl kind: Runtimeclass
metadata:
  name: ""
handler: ""
candidate@cli:~$ vim /home/candidate/KSMV00301/runtime-class.yaml
candidate@cli:~$ cat /home/candidate/KSMV00301/runtime-class.yaml

apiVersion: node.k8s.io/vl kind: Runtimeclass metadata: name: "sandboxed"
handler: "runsc"
candidate0cli:~$ kubectl get deployments.apps -n server
NAME        READY UP-TO-DATE AVAILABLE       AGE
workloadl 1/1       1           1          5h43m
workload2 1/1       1           1          5h43m
workload3 1/1       1           1          5h43m
candidate@cli:~$ kubectl get pods -       -n
NAME                         READY    STATUS    RESTARTS   AGE
workloadl-6869857dd7-s45rc   1/1       Running  0          5h43m
workload2-d4bd497d5-h44df    1/1       Running  0          5h43m
workload3-8587774495-chm56   1/1       Running  0          5h43m
candidate@cli:~$ kubectl -n  server edit deployments.appsworkloadl |
```

www.atmicnetworks.com

```
  template:
    metadata: creationTimestamp: null labels:
          i •: nginx workloadl
    spec:
                    <       : sandboxed
      containers:
        -       - nginx:1.14.2
                          ItNotPresent workloadl
        ports:
        - containerPort: 1
          p > ' 1: TCP resources: {}
                                  /dev/termination-log
                                                File
                ClusterFirst |     Always
                    default-scheduler    securityContext:    I}
terminationGracePeriodSeconds: 30 status:
"/tmp/kubectl-edit-3385772700.yaml"
```

```
NAME                          READY    STATUS     RESTARTS    AGE
workloadl-6869857dd7-s45rc    1/1      Running    0           5h44m
worklcad2-d4bd497d5-h44df     1/1      Running    0           5h44m
workload3-8587774495-chm56    1/1      Running    0           5h44m
candidate@cli:~$ kubectl -n server Edit cancelled,  edit deployments.apps        workloadl
no changes made. candidate@cli:~$ kubectl get
pods -                                -n server
NAME                          READY    STATUS     RESTARTS    AGE
workloadl6869857dd7 s45rc     1/1      Running    0           5h45m
workload?-d 4bd 4 97 d 5-h 4 4 d f    1/1      Running    0           5h44m
workload3-B587774495-chm56    1/1      Running    0           5h44tn
candidate@cli:~$ kubectl -n   server    edit deployments.apps        workload?
 Edit cancelled, no changes made.
candidate?cli:~$ kubectl create -f /home/cand.idate/KSMV00301/runtime-class.yaml
runtimeclass.node.k9s.io/sandboxed created
                              pods -      -n server
candidate@cli:-$ kubectl get NAME    READY    STATUS     RESTARTS    AGE
workloadl-6869857dd7-s45rc    1/1      Running    0           5h45m
workload?-d4bd497d5-h44df     1/1      Running    0           5h45m
workload3-8587774495-chm56    1/1      Running    0           5h45m
candidate@cli:-? kubectl -n :    server      edit deployments.apps workload?
                              25%
                                  25%
                RollingUpdate
```

: nginx

workload2

sandboxed

```
NAME                          READY     STATUS    RESTARTS AGE
workloadl-6869857dd7-s45rc    1/1       Running   0          5h45m
workload2-d4bd497d5-h44df     1/1       Running   0          5h45m
workload3 8587774495 chm56    1/1       Running   0          5h45m
candidate@cli:~$ kubectl -n server edit deployments.apps workload2
deployment.apps/workload2 edited candidate@cli:~$ kubectl get  pods
                              -n server
NAME                          READY     STATUS    RESTARTS AGE
workload!-8d8649ff6-wvjtg     1/1       Running   0          15s
workload2-765bdb98c8-wd8cm    1/1       Running   0          4s
workload3-8587774495-chm56    1/1       Running   0          5h45m
candidate@cli:~$ kubectl -n server edit deployments.apps workload3
```

1

nginx workload3

sandboxed

nginx:1.14.2

IfNotPresent

workloads

ports: _____

```
candidate@cli:~$ kubectl -n server edit deployments.apps workload3
deployment.apps/workload3 edited

candidate@cli:~$ kubectl get pods -n server
NAME                         READY   STATUS    RESTARTS   AGE
workloadl-8d8649ff6-wvjtg    1/1     Running   0          58s
workload2-765bdb98c8-wd8cm   1/1     Running   0          47s
Workload3-76c994bb4d-s6k85   1/1     Running   0          4s
candidate@cli:~$ f
```

Question: 43
SIMULATION

www.atmicnetworks.com

You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSCHOO	ksch00301	ksch00301 301 -master -workerl

You can switch the cluster/configuration context using the following command:

```
[candidatqfcli] $ kubec tl config use-context KS CH00301
```

Context
Your organization's security policy includes:

ServiceAccounts must not automount API credentials

www.atmicnetworks.com

ServiceAccount names must end in "-sa"

The Pod specified in the manifest file /home/candidate/KSCH00301 /pod-m

nifest.yaml fails to schedule because of an incorrectly specified ServiceAccount.

Complete the following tasks:

Task

1. Create a new ServiceAccount named frontend-sa in the existing namespace qa. Ensure the ServiceAccount does not automount API credentials.

2. Using the manifest file at /home/candidate/KSCH00301 /pod-manifest.yaml, create the Pod.

3. Finally, clean up any unused ServiceAccounts in namespace qa.

Answer: See the explanation below

Explanation:

www.atmicnetworks.com

```
Switched to context "KSCH00301". candidate@cli:~$ kubectl get sa -n qa

NAME         SECRETS    AGE
default      1          5h46m
podrunner    1          5h46m
candidate@cli:~$ kubectl get deployment -n qa No resources found in qa
namespace. candidate@cli:~$ kubectl get pod -n qa No resources found in qa
namespace.
candidate@cli:~$ kubectl create sa frontend-sa -n qa serviceaccount/frontend-
sa created candidate@cli:~$ kubectl get sa -n qa

NAME           SECRETS    AGE
default        1          5h47m
frontend-sa    1          4s
podrunner      1          5h47m
candidate@cli:~$ cat /home/candidate/KSCH00301/pod-manifest.yaml apiVersion:
vl kind: Pod metadata: name: "frontend" namespace: "qa" spec:
serviceAccountName: "frontend-sa" containers: - name: "frontend" image: nginx
candidate@cli:~$ vim /home/candidate/KSCH00301/pod-manifest.yamll
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: "frontend"
  namespace: "qa"
spec:
  serviceAccountName: "frontend-sa"
  automountServiceAccountToken: false
  containers:
    - name: "frontend"
      image: nginx
```

```
candidate0cli:~$ vim /home/candidate/KSCH0Q301/pod-manifest.yaml candidate@cli:~$ cat
/home/candidate/KSCH00301/pod-manifest.yaml apjvers ion: vl kind: Pod metadata: name: "frontend"
namespace: "qa" spec: serviceAccountName: "frontend-sa" automountServiceAccountToken: false containers:
    - name: "Erontend" image: nginx candidate@cli:~$ kubectl create -f /home/candidate/KSCH00301/pod-
manifest.yaml pod/fcontend created candidate^cli:-$ kubectl get pods -n qa NAME          READY STATUS
          RESTARTS          AGE
frontend    1/1       Running    0          6s
candidate0cll:~$ kubectl get sa -n qa NAME SECRETS AGE
default       1          5h49m
frontend-sa   1          105s
podrunner     1          5h49m
candidate0cli:-$ kubectl delete sa/podrunner -n qa serviceaccount "podrunner" deleted candidateScli:~$ Q
```

www.atmicnetworks.com

Question: 44

SIMULATION

www.atmicnetworks.com

You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSCS002	kscs00201	kscs00201
01	-master	-workerl

You can switch the cluster/configuration context using the following command:

```
[ candidate^ li] $ kubec tl config use-context KS
CS00201
```

Context

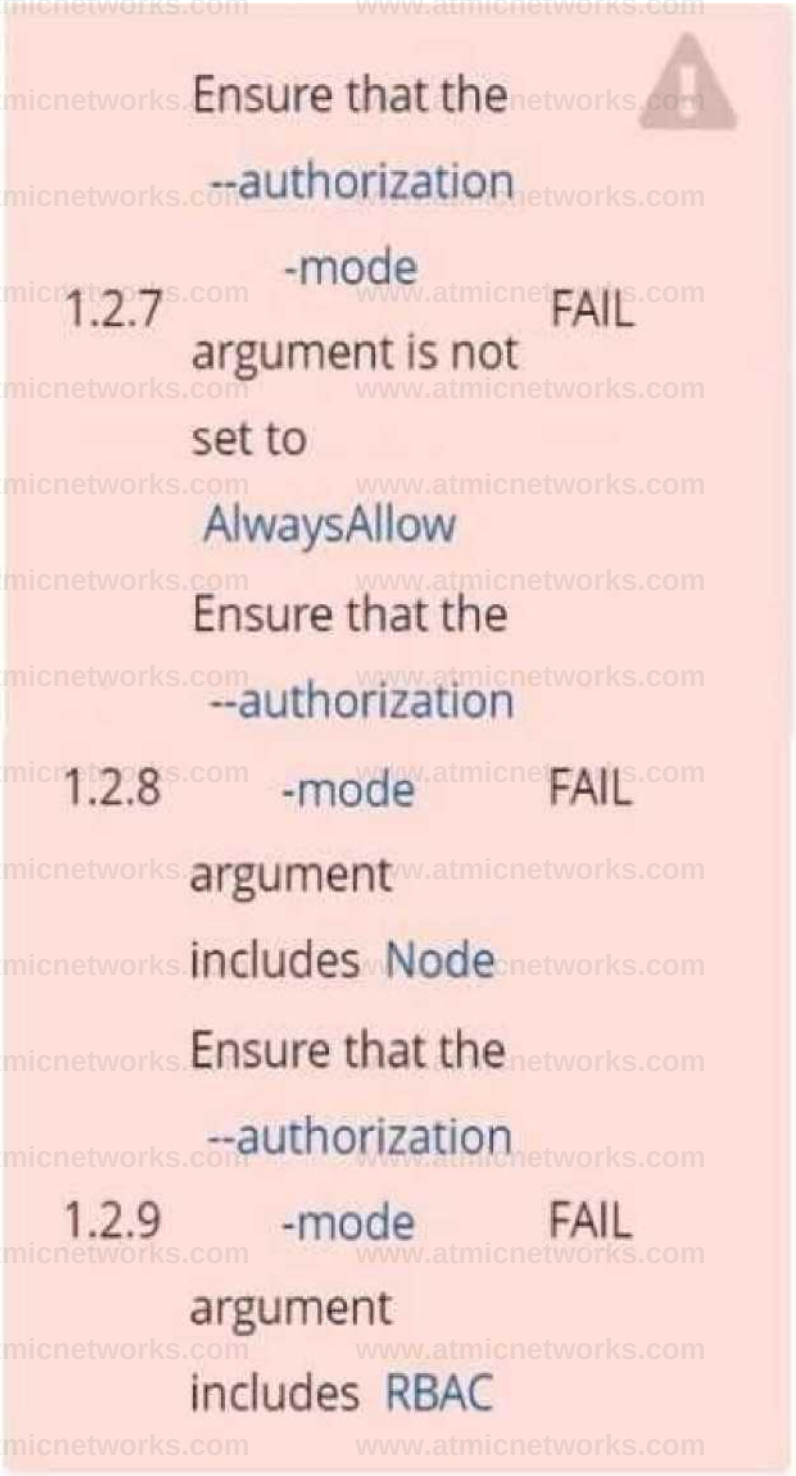
A CIS Benchmark tool was run against the kubeadm-created cluster and found multiple issues that must be addressed immediately.

Task

Fix all issues via configuration and restart the affected components to ensure the new settings take effect.

www.atmicnetworks.com

Fix all of the following violations that were found against the API server:



Fix all of the following violations that were found against the Kubelet:

www.atmicnetworks.com

Ensure that the anonymous-auth FAIL argument is set to false

4.2.1

Ensure that the -authorization-mode argument is not set to AlwaysAllow

4.2.2

Use Webhook authentication/authorization where possible.

Fix all of the following violations that were found against etcd:

2.2	Ensure that the —client-cert-auth argument is set to true	FAIL

Answer: See explanation below.

Explanation:

www.atmicnetworks.com

www.atmicnetworks.com

```
candidate@cli:-$ kubectl delete sa/podrunner -n qa
service-account "padrunner" deleted
candidate@cli:~$ kubectl config use-context KSCS00201
Switched to context "KSCS0O2O1".
candida te@d i: ~$ ssh kscs00201-master
Warning: Permanently added '10.240.86.194' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in
the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

root8kscs00201-master:~l vim /etc/kubernetes/manifests/kube-apiserver.yaml
rootskscs00201-master:-I systemctl daemon-reload
root@kscs00201-master:~* systemctl restart kubelet.service
root@kscs00201-master:'* systemctl enable kubelet.service
root@kscs00201-master:-t systemctl status kubelet.service^
• kubelet.service - kubelet: The Kubernetes Node Agent
     Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled) Drop-In:
/etc/systemd/system/kubelet.service.d
             •—10-kubeadm.conf
     Active: active (running) since Eri 2022-05-20 14:19:31 UTC; 29s ago Docs: https://kubernetes.io/docs/home/
   Main PID: 134205 (kubelet)
      Tasks: 16 (limit: 76200)
     Memory: 39.5M
     CGroup: /system.slice/kubelet.service
             *—134205 /usr/bin/kubelet —bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubj

May 20 14:19:35 kscs00201~master kubelet[134205]: May 20 10520 14:19:35.420825 134205 reconciler.
14:19:35 kscs00201-master kubelet[134205]: May 20 14:19:35 10520 14:19:35.420863 134205 reconciler.
kscs00201 roaster kubelet[134205]: May 20 14:19:35 kscs00201- 10520 14:19:35.420907 134205 reconciler.
master kubelet[134205]: May 20 14:19:36 kscs00201-master 10520 14:19:35.420928 134205 reconciler.
kubelet[134205]: May 20 14:19:37 kscs00201-master 10520 14:19:36.572353 134205 request.go:
kubelet[134205]: May 20 14:19:37 kscs00201-master 10520 14:19:37.112347 134205 prober mana
kubelet[134205]: May 20 14:19:37 kscs00201-master E0520 14:19:37.185076 134205 kubelet.go:
kubelet[134205]: May 20 14:19:38 kscs00201-master 10520 14:19:37.645798 134205 kubelet.go:
kubelet[134205]: May 20 14:19:40 kscs00201 master 10520 14:19:38.184062 134205 kubelet.go:
kubelet[134205]:                                  10520 14:19:40.036042 134205 probermana
limes 1-22/22 (END)

de Agent
et.service; enabled; vendor preset: enabled) ce.d

5-20 14:19:31 UTC; 29s ago



trap-kubeconfig=/etc/kubernetes/bootstrap-kubelet,conf —kubeconfig=/etc/kubernetes/kubeler]

5]: 10520 14:19:35.420825 5): 134205 reconciler.go:221] "operationExecutor.VerifyControllerAt.tB
10520 14:19:35.420863 5J: 134205 reconciler .go: 221 ] "iperationExecutor.Veri fyControllerAtJ
10520 14:19:35.420907 5[: 134205 reconciler.go:221) "operationExecutor.VerifyControllerAtJ
10520 14:19:35.420928 5]: 134205 reconciler.go:157) "Reconciler: start to sync state"
10520 14:19:36.572353 5]: 134205 request.go:665] Waited for 1.049946364s due to client-sic
10520 14:19:37.112347 5]: 134205 prober_manager.go:255] "Failed to trigger a manual run" f
E0520 14:19:37.185076 5]: 134205 kubelet.go:1711] "Failed creating a mirror pod for" err-"
10520 14:19:37.645798 5]: 134205 kubelet.go:1693) "Trying to delete pod" pod="kube-system/
10520 14:19:38.184062 5]: 134205 kubelet.go:1698j "Deleted mirror pod because it is outdat
10520 14:19:40.036042 134205 prcber_manager.go:255] "Failed to trigger a manual run" p

lines 1-22/22 (END)
let.conf —kubeconfig^/etc/kubernetes/kubelet .conf —config=/var/lib/kubelet/config.yaml —]
0:221] "operationExecutor.VerifyControllerAttachedVoltune started for volume \"kube-proxy!" I o:221]
"operationExecutor.VerifyControllerAttachedVolume started for volume \"lib-modules\"l o:221]
"operationExecutor.VerifyControllerAttachedVolume started for volume \"flannel-cfg\"| 6:157] "Reconciler: start to sync state"
65] Waited for 1.049946364s due to client-side throttling, not priority and fairness, reguej er.go:255] "Failed to trigger a manual
```

www.atmicnetworks.com

```
run" probe="Readiness"
711] "Failed creating a mirror pod for" err="pods \"kube-apiserver-kscs00201-master\" alreal 693] "Trying to delete pod"
pod="kube-system/kube-apiserver~kscs00201-master" podUID=bb91ell 69S] "Deleted mirror pod because it is outdated"
pod="kubc system/kube apiserver kscsOT201 I er.go:255] "Failed to trigger a manual run" probe="Readiness"
```

```
root@kscs00201-master:~# vim /var/lib/kubelet/config.yaml I
```

```
apiVersion: kubelet.config.k8s.io/v1beta1
authentication:
  anonymous:
    enabled: false
  webhook:
    cacheTTL: 0s
    enabled: true
  x509:
    clientCAFile: /etc/kubernetes/pki/ca.crt
authorization:
  mode: Webhook
  webhook:
    cacheAuthorizedTTL: 0s
    cacheUnauthorizedTTL: 0s
cgroupDriver: systemd
clusterDNS:
```

```
root@kscs00201-master:~# vim /var/lib/kubelet/config.yaml
root0kscsOO2Ol-master:~i vim /var/lib/kubelet/config.yaml
root@kscs00201-master:~l vim /etc/kubernetes/manifests/eted.yaml root@kscs00201-master:~l systemctl daemon-reload
root@kscs00201-master:~f systemctl restart kubelet.service root@kscs00201-master:~i systemctl status kubelet.service!
```

www.atmicnetworks.com

```
• kubelet.service - kubelet: The Kubernetes Node Agent
     Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
    Drop-In: /etc/systemd/system/kubelet.service.d
             [1]—10-kubeadm.conf
     Active: active (running) since Fri 2022-05-20 14:22:29 UTC; 4s ago
       Docs: https://kubernetes.io/docs/home/
   Main PID: 135049 (kubelet)                    J^
      Tasks: 17 (limit: 76200)
     Memory: 38.0M
     CGroup: /system.slice/kubelet.service
             •—135049 /usr/bin/kubelet                —bootstrap-kubeconf ig=/etc/kube metes/bootstrap-kulJ

May 20 14:22:30 kscs00201-master kubelet[135049]:
May 20 14:22:30 kscs00201-master kubelet[135049]:
May 20 14:22:30 kscs00201 master kubelet[135049]:
May 20 14:22:30 kscs00201-master kubelet[135849]:
May 20 14:22:30 kscs00201-master kubelet[135849]:
May 20 14:22:30 kscs00201-master kubelet[135849]:
May 20 14:22:30 kscs00201 master kubelet[135049]:
May 20 14:22:30 kscs00201-master kubelet[135849]:
May 20 14:22:30 kscs00201-master kubelet[135849]:
May 20 14:22:30 kscs00201-master kubelet[135849]:
lines 1-22/22 (END)

10520 14:22:30.330232    135849   reconciler.
10520 14:22:30.330259    135849   reconciler.
10520 14:22:30.330304    135849   reconciler.
10520 14:22:30.330354    135849   reconciler.
10520 14:22:30.330378    135849   reconciler.
10520 14:22:30.330397    135849   reconciler.
10520 14:22:30.330415    135849   reconciler.
10520 14:22:30.330433    135849   reconciler.
10520 14:22:30.330452    135849   reconciler.
10520 14:22:30.330463    135849   reconciler.

May 20 14:22:30 kscs00201-master kubelet[135849]: 10520 14:22:30.330463 135849 reconciler.| root@kscs00201-master:'*
roots kscs00201 -master: •-# root@kscs00201-master:'* root@kscs00201-master:^f exit logout
Connection to 10.240.86.194 closed.
candidate@cli:'$ |
```

Question: 45

SIMULATION

www.atmicnetworks.com

You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSCS001	kscsOO1O1 01 -master	kscsOO1O1 -workerl

You can switch the cluster/configuration context using the following command:

```
[candidate@cli] $ kubec tl config use-context KS CS00101
```

Context

A default-deny NetworkPolicy avoids to accidentally expose a Pod in a namespace that doesn't have any other NetworkPolicy defined.

Task

Create a new default-deny NetworkPolicy named defaultdeny in the namespace testing for all traffic of type Egress.

The new NetworkPolicy must deny all Egress traffic in the namespace testing.

Apply the newly created default-deny NetworkPolicy to all Pods running in namespace testing.

You can find a skeleton manifest file at /home/candidate/KSCS00101 /network-policy.yaml

Answer: See explanation below.

Explanation:

```
candidate@cli:~$ kubectl config use-context KSCS00101 Switched to context
"KSCS00101".
candidate0cli:-$ cat /home/candidate/KSCSOOlOl/network-policy.yaml

apiVersion: networking.k8s.io/vl
kind: Networkpolicy
metadata:
   name: ""
   namespace: ""
spec:
   podSelector: ()
   policyTypes: []
candidate@cli:~$ vim /home/candidate/KSCSOOlOl/network-policy.yaml
candidate@cli:~$ |
```

www.atmicnetworks.com

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: "defaultdeny"
  namespace: "testing"
spec:
  podSelector: {}
  policyTypes:
  - Egress
  egress:
  - to:
    - podSelector: {}
      namespaceSelector:
        matchLabels:
          access: testingproject
```

```
candidate?c1i:-$ vim /home/candidate/KSCSOO101 /network-policy.yaml candida teSdi: ~$ vim
/home/candidate/KSCS00101/network-policy.yaml candidate@cli:~$ kubectl label ns testing access=testingproject
namespace/test!ng labeled candidate@cli:~$ cat /home/candidate/KSCS00101/network-policy.yaml apiversion:
networking.k8s.io/vl kind: Networkpolicy metadata: name: "defat: ltdeny" namespace: "testing" spec: podSelector: {} policyTypes:
- Egress egress: - to: - podselector: () namespaceSelector: matchLabels: access: testingproject
candidateScli:~$ kubectl create -f /home/candidate/KSCSOOlOl/network-policy.yaml
networkpolicy.networking.k8s.io/defaultdeny created candidate@cii:~5 kubectl -n testing describe networkpolicy Name:
                defaultdeny
Namespace:      testing
Created on: 2022-05-20 14:28:27 +0000 UTC Labels:           <none>
Annotations: <none> Spec:
  PodSelector:          <none> (Allowing the specific traffic to all pods in this namespace}
  Not affecting ingress traffic Allowing egress traffic: To Port: <any> (traffic allowed to all ports) To: NamespaceSelector:
access=testingproject PodSelector: <none> Policy Types: Egress candidate@cli:~$ |
```

Question: 46

SIMULATION

You can switch the cluster/configuration context using the following command:

[candidate^ li] $ kubec tl config use-context KS MV00102

www.atmicnetworks.com

Context

A PodSecurityPolicy shall prevent the creation of privileged Pods in a specific namespace.

Task

Create a new PodSecurityPolicy named prevent-psp-policy,which prevents the creation of privileged Pods.

Create a new ClusterRole named restrict-access-role, which uses the newly created PodSecurityPolicy prevent-psp-policy.

Create a new ServiceAccount named psp-restrict-sa in the existing namespace staging.

Finally, create a new ClusterRoleBinding named restrict-access-bind, which binds the newly created ClusterRole restrict-access-role to the newly created ServiceAccount psp-restrict-sa.

You can find skeleton manifest files at:

- /home/candidate/KSMVOO 102/pod-security-policy.yaml
- /home/candidate/KSMVOO 102/cluster-role.yaml
- /home/candidate/KSMVOO 102/service-account.yaml
- /home/candidate/KSMVOO 102/cluster-role-bindmg.ya

www.atmicnetworks.com

ml

Answer: See explanation below.

Explanation:

```
candidate@cli:-$ kubectl config use-context KSMV00102 Switched to context
"KSMV00102".
candidate?cli:~$ cat /home/candidate/KSMV00102/pod-security-poiicy.yaml

apiVersion: policy/vlbetal kind: PodSecurityPolicy metadata: name: ""

spec:
    seLinux: rule: "" runAsUser: rule: ""
    supplementsIGroups: {} fsGroup: {}
candidate^cli:~$ vim /home/candida te/KSMVOO102/pod-securi t y-policy.yaml!
```

policy/vibetai PodSecurityPolicy metadata:

RunAsAny

RunAsAny

RunAsAny

RunAsAny

```
candidate@cli:~$ vim /home/candidate/KSMV00102/pod-security-policy.yamJ
candidate@cli:~? cat /home/candidate/KSMV00102/pod—security—policy.yaml

apiVersion: policy/vlbetal kind: PodSecurityPolicy metadata:
    name: "prevent-psp-policy" spec:
    privileged: false seLinux:
        rule: RunAsAny runAsUser:
        rule: RunAsAny

    supplementalGroups: rule: RunAsAny
```

www.atmicnetworks.com

```
    fsGroup:

        rule: RunAsAny
candidate@cli:~$ kubectl create -f /home/candidate/KSMV00102/pod-security-policy.yaml
Warning: policy/vlbetal PodSecurityPolicy is deprecated in vl.21+, unavailable in vl.25+ podsecuritypolicy.policy/prevent-psp-policy created
candidate@cli:'$ cat /home/candidate/KSMV00102/cluster-role.yaml

apiVersion: rbac.authorization.k8s.io/vl kind: ClusterRole metadata: name: "" rules:
candidate@cli:~$ vim /home/candidate/KSMV00102/cluster-role.yaml|
```

rbac.authorization.k8s.io/vl

_ ClusterRole metaoata •

```
candidate@cli:~5 kubectl create clusterroie restrict-access-role —verb=use —resource-psp - -dry-runnel lent -a yaml
apiVersion: rbac.authorization.k3s.io/vl kind: ClusterRole metadata:
    creationTimestamp: null
    name: restrict-access-role
rules:
- apiGroups:
    - policy
    resources:
    - podsecuritypolicies
    verbs:
    - use
candidate@cli:~5 vim /home/candidate/KSMV00102/cluster-role.yaml]
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: "restrict-access-role"
rules:
- apiGroups:
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
candidate@cli:~$ vim /home/candidate/KSMV0D102/cluster-role.yaml
candldate@cli:~$ kubectl create clusterrole restrict-access-role —verb=use —resource=psp -

-dry-run-client —resource-name-prevent-psp-pol±cy -o yaml apiversion: rbac.authorization.IBs.io/v] kind: ClusterRole metadata;
    creationTimestamp: null name: restrict-access-role rules: - apiGroups:
    - policy resourceNames: - prevent-psp-policy resources:
```

```
  - podsecuritypolicies
  verbs:

  - use
candidate0cll:~$ vim /home/candidate/KSMV00102/cluster-role.yaml|
```

```
rbac.author!zation.k8s.io/vl ClusterRole
metadata:
```

```
rules:
- apiGroups:
```

```
      policy

  resourceNames:
    - prevent-psp-policy resources:
      podsecuritypolicies verbs:
    - use
candidate@cli:~$ kubectl create -f /home/candidate/KSMV00102/cluster~role.yaml
clusterrole.rbac.authorization.k0s.io/restrict-access-role created candidate@cli:~$ candidate@cli:~$
candidate@cli:~$ cat /home/candidate/KSMV00102/service-account.yaml I
```

```
metadata:
  name: "j : { - re ^ r i rt a"
  name space: "staging"
```

```
apiVersion: vl
kind: ServiceAccount
metadata:
    name: " "■
    namespace: ""
candidateBcli:~S vim /home/candidate/KSMV00102/service account.yaml
candidate0cli:~$ cat /home/candidate/KSMV00102/service account.yaml

apiVersion: vl
kind: ServiceAccount
metadata:
  name: "psp-restrict-sa" namespace: "staging"
candidate@cli:-5 kubectl get sa -n staging

NAME      SECRETS AGE
default   1       6h6m
candidateBcll:-$ kubectl create -f /home/candidate/KSMV00102/service-account.yamlj serviceaccount/psp-restrict-sa created
candidate@cli:~$ kubectl get sa -n staging
NAME              SECRETS  AGE
default           1        6h6m
psp-restrict-sa   1        2s
candidateBcii:-5
candidateBcli: ~$
```

```
candidate®cll:~$ kubectl create clusterrolebinding restrict-access-blnd —clusterrole=restri ct-access-role —
serviceaccount=staging:psp-restrict-sa --dry-run -o yaml
W0520 14:41:23.502004        47621 helpers.go:598] —dry-run is deprecated and can be replaced wi
th —dry-run=client.
apiVersion: rbac.author!zation. k8 s. io/vl
kind: ClusterRoleBinding
metadata:
  creationTimestamp: null name: restrict-aocess-bind
roleRef:

  apiGroup: rbac.authorization.k8s.io

  kind: ClusterRole name: restrict access■role subjects: - kind: ServiceAccount
  name: psp-restrict-sa namespace: staging
candidateBcii:-? vim /home/candidate/KSMV00102/cluster-role


cluster-role-binding.yaml cluster-role.yaml
candidate0cli:-S vim /home/candidate/KSMV00102/cluster-role
cluster-role-bindlng.yaml cluster-role.yaml
candidateBcii:~$ vim /home/candidate/KSMV00102/cluster-role-binding.yami |
```

rbac.authorization. k8s.io/vl ClusterRoleBinding

restrict-access-bind

rbac.authorization.k8s.io ClusterRole restrict access-role

1: ServiceAccount psp-restrict-sa staging

```
apiVersion: rbac.authorization. k3s.io/vl kind: ClusterRoleBinding metadata:
  name: restrict-access-bind roleRef:
  apiGroup: rbac.authorization.kBs.io kind: ClusterRole name: restrict-access-role
subjects:
- kind: ServiceAccount name: psp-restrict-sa namespace: staging
```

```
candidate@cli:~$

candidate@cli:~$ kubectl create -f /home/candidate/KSMV00102/cluster-role-binding.yaml
clusterrolebinding.rbac.authorization.kBs.io/restrict-access-bind created candidategcli:~$ Q
```

Question: 47
SIMULATION

You **must** complete this task on the following cluster/nodes:

Master Worker

Cluster node node

KSSC003 kssc00301

kssc00301

01 -master

-workerl

You can switch the cluster/configuration context using the following command:

[candidate^ li] $ kubec tl config use-context KS SC00301

Task

Analyze and edit the given Dockerfile /home/candidate/KSSC00301/Docker file (based on the ubuntu:16.04 image), fixing two instructions present in the file that are prominent security/best- practice issues.

www.atmicnetworks.com

Analyze and edit the given manifest file /home/candidate/KSSC00301/deployment.yaml, fixing two fields present in the file that are prominent security/best-practice issues.

Don't add or remove configuration settings; only modify the existing configuration settings, so that **two** configuration settings each are no longer security/best- practice concerns.

Should you need an unprivileged user for any of the tasks, use user nobody with user id 65535.

Answer: See explanation below.

Explanation:

```
 candidate@cli:-$ kubectl config use-context KSSC00301
Switched to context "KSSC00301".
candidate@cli:~$ vim KSSC00301/Dockerfile
```

```
FROM ubuntu:16.04|

USER root
RUN apt-get update SS \
        apt-get install -yq —no-insLall-recommends runiti=2.L.2-3ubuntul wget=l.17.1-lubuntul.5

            chrpath=0.16-1 tzdata=2020a-0ubuntu0.16.04 lsof-4.S9+dfsg-0.1 lshw=02.17-1.Iubuntu3

            syssLat=ll.2.O-lubuntuO.3 net-tools=l,60-26ubuntul numactl=2.0.11-lubuntul.1 \ bzip2=1.0.6'-8ubuntu0.2 ss \
        apt-get autorenove ss apt-get clean ss \


        rm -rf /var/lib/apt/lists/* /tap/* /var/tmp/*
ARG CB_VERSION=6.5.1
AR<, CB_RELEASE_URL=https://packages.couchbase.com/releases/6.5.1
ARC CB_PACKAGE=couchb:...e-server-enterprise_6.5. l-ubuntul6.04_amd64.deb
ARG CB_SHA256=80427193137e5cb5a4795b2675blc450claf8cfla5c634d917f6c416f2047e66

KU PATH-SPATH:/opt/couchb- -e/bin:/opt/couchbase/bin/tools:/opt/couchb e/bin/i nstall
```

```
RUN groupadd -g 1000 couchbare SS useradd couchbve u 1000 -g couchb e M

SHELL [                ,       ,                ,        "]
RUN export INSTALL DONT START SERVER-1 SS \
        wqct N no verbose $CB_RELE.P E_UR.L/?CB PACKAGE SS \
        echo                                    I 3ha256sum -c - ss \
        dpkg -1 ./$CB_PACKAGE SS rm -f ./SCB_?ACKAGE
COPY scr+pts/run /ctc/rcrvice/couchbare-scrver/run
RUN chown -R couchb ace:couchba. e /etc/service

COPY scripts/dummy.sh /usr/local/bin/
RUN In -s dummy.sh. /usr/local/bin/iptables-save SS \
        In -s dummy.sh /usr/local/bin/lvdisplay SS \
        In -s dummy.sh /usr/local/bin/vgdisplay ss \
        In -s dummy.sh /usr/local/bin/pvdisp.lay

RUN chrpath -t                          /opt/couchbce/bin/curl

COPY scripts/entrypolnt.sh /
ENTRYPOINT [                    ]
USER nobody
CMD [                           ]

EXPOSE 8091 8092 8093 8094 8095 8096 11207 11210 11211 18091 18092 18093 18094 18095 18096
VOLUME /opt/couchbase/var
```

```
 candidate@cli:~$ kubectl config use-context KSSC00301
Switched to context "KSSC00301".
candidate@cli:~$ vim KSSC00301/Dockerfile
candidate@cli:~$ vim KSSC00301/deployment.yarn! |^
```

Question: 48
SIMULATION

www.atmicnetworks.com

You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSCHOO	kschOO1O1 101 -master	kschOO1O1 -workerl

You can switch the cluster/configuration context using the following command:

[candidate^: S kubec tl config use-context KS CH00101

Context

The kubeadm-created cluster's Kubernetes API server was, for testing purposes, temporarily configured to allow unauthenticated and unauthorized access granting the anonymous user dusteradmin access.

Task

Reconfigure the cluster's Kubernetes API server to ensure that only authenticated and authorized REST requests are allowed.

Use authorization mode Node,RBAC and admission controller NodeRestriction.

Cleaning up, remove the ClusterRoleBinding for user system:anonymous.

All kubectl configuration contexts/files were also configured to use the unauthenticated and unauthorized access. You don't have to change that, but be aware that kubectl's configuration will stop working, once you've completed securing the cluster.

You can use the cluster's original kubectl configuration file /etc/kubernetes/admin.conf, located on the cluster's master node, to ensure that authenticated and authorized requests are still allowed.

Answer: See explanation below.

Explanation:

```
candidate@cli:-$ kubectl config use-context KSCH00101
Switched to context "KSCH00101".
candidate@cli:~$ ssh kschO0101-master
Warning: Permanently added '10.240.86.190' (ECDSA) to the list of known hosts.
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the individual files in
/usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml I
```

```
B                    Vl
          Pod
metadata:

  annotations:
                                                    10.240.86.190:6443
  creationTimestamp: null labels:
```

www.atmicnetworks.com

```
    1:   kube-apiserver control-plane
kube-apiserver
kube-system
spec: containers: — command:
-   kube-apiserver - —advertise-address—10.240.86.190 - —allow-privileqed= ■ —authorization-mode=Node,RBAC
■client-ca-file=/etc/kubernetes/pki/ca.crt
-   —enable-admission-plugins=AiwaysAdmit
-   --enable-bootstrap-token-auth-
—etcd“cafile=/etc/kubernetes/pki/etcd/ca.crt
-   —etcd-certfile-/etc/kubernetes/pki/apiserver-etcd-cllent.crt
-   —etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd“client.key "/etc/kubernetes/manifests/kube-apiserver.yaml"
128L, 4343C1,1                                                                                    Top
```

www.atmicnetworks.com

```
rootSkschOOlOl-master:'I cat /etc/kubernetes/admin.conf apiVersion: vl clusters:
- cluster:
certificate-authority data: LS0tLSlCRUdJTiBDRVJUSUZJQ0FURS0tLS0tCklJSUMvakNDQWVhZ0F3SUJB
Z01CQURBTkJna3Foa21HOXcwQkFRcOZBREFWTVJNdOVRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJ
eUlESXhO
akF3TlRVeE9WblhEVE15TURJeE5EQXdOVFV4TlZvdOZURVRNQkVHQTFVRQpBeElLYTNWaVpYSnVaWFJsY3pDQOF
TSXdE
UVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTlgwCm9LeUYvTGNmYTIvNzNZTktkSFdZU3JUaUx0
QStr N01qTXpRZllzM2ttNGllalpoM0tZc3YlbUdpN0UyQ2tYc0MKUnhlLlNiZnBDMzUa2k5V3h0SHc5eTM0OEtXUVE3VXBL
UmZRdXVxdlAlWXdDZkordlJmWGNGTXQxLzRNQVhWLwpkdjZ5YWRKSitPeFFSVjZlaHFBZHROM3FtOFdVcW84UE5J
TlEO
OEc3WWhnRUg5RHU3SFdkMS8raXVkSjNOMX16CnNISEdtYklsWENSbEcydFVOM2RScDczSnRISUJS2tnMGxYM3FWSl
Uy
QmJRblBmK01wb0VlTXFGanZVcWVaVWcKYlBKK3ROVmZIMlJLTkhVUnYydVJIa3ZZc2JrclhUMW8rMXFNNHZrYnFN
MHlq
KzNxTUtiSyt5V3dzUTlBYUVPMApUdXR4UUdlTFp3OUE3TjZZeTFVQ0F3RUFBYU5aTUZJd0RnWURWUjBQQVFIL0JBU
URB
ZOtrTUE4ROExVWRl'dOVCCi93UUZNQUlCQWY4dOhRWURWUjBPQkJZRUZEcUiwLzdYbzZaNkJNViVEK2w3bFZPcGp
BOWlN
QlVHQTFVZEVRUU8KTOF5Q0NtdDFZbVZ5YmlWMFpYTXdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBSlNWNm9w
NGgxYkNv
eGZLRUZ4bwoxaVlHUF1nMlhhOTNOWEZlTTY3RnA2NkdqUEc5SXBONnNHUnRnWVlydOMya1BDeFVOb2IySWtUQlFN
btW3
cWRHCkdPS2JwWp6Smc3Y0dyS2E3RlpZWVNyVUVGRWhyd2xZWXNGME56aFBoZVcwcHJjcWtSdXNlbm55SG5YNGVO
MUoK
NlNzbGZYTjJldVFJdlVIRG15LOJsLlZWRmZNZnRxOGFOZOpYSFZGTmlVcDRpNXlJTXFRNTB4ZjVqcnFlWFRmVwpVd
mJq
ZjEyOThXVTk3QkxHcDdRZE9QYWVKU051UStlVkMrdnpVZ2tVQVNjclVsc24xcThPNnBRbjV3TjNxdUVrCm5zQk9pckxS
c2k2alN3UlhIbGcvangvcitqdOdTcOxwWUxDZTlxalFraTdCSVRJTlN3ejd3c2hzbF.RuNzBFYOIKaOVBPQotLSOtLUVO
RCBDRVJUSUZJQOFURSOtLSOtCg-
server: https://10.240.86.190:6443
name: kubernetes contexts:
- context:
cluster: kubernetes
user: kubernetes-admin
name: kubernetes-adminSkubcrnetes
current-context: kubernetes-admin@kubemetes kind: Config preferences: () users: - name: kubernetes-admin
user:
client-certificate-data: LS0tLSlCRUdJriBDRVJUSUZJQ0FURS0tLS0tCklJSURJVENDQWdtZ0F3SUJBZ01
ocEdQcDB4Zk9JbkYxaGJwcTh5YlBUMGxlTm5VNjBiSUpxRXVKckxJbEtXClNValhlVkYzNklOZHclZU10T2JxKlhaaHd
hY2JURVZCMlVDVURsbDgzdG5teFQyVXJmY0pUQm!iLTCtZTFAvcWYKdjdXR3BwQlZXNnhVZGFibGNuUklMnpleUVJ
TEt
TckSXbUQ0TzZsMU13blZ0OVJzQ2RXTkV3VGNZRHdoUTd2OQpGcExKL3hiSDdUTzkwYlRFdlIwazl3cFVYdllkdkljSXN
MRkYwL3F2bDA3U31xbGplOEIlSnNpQlhCUlZxbS9wCmNlRJSs3SnZlbtndaZzlkOWdZaVJVdFFTcHBONkx4UnhkSzNKMG
R
BK240SWxFZEtHRWh3TEO0d0tHaldERG9scHgKYzB3WHkwVXBORGZ6OUxuRUFzVUJsbDRCQ3VkdW5QNWDN2FuS3
dJREF
RQUJBb01CQURWRkZNSVRqYnNySTZTTwpQOGHOMTByN3RWZ251cXJVS202dHRnZWtXOWdlSlpvMnZyb3RsbG9qO
GFRamF
0MTZnaEUwOXdZd2xMSDhId0tLCklMb2NrZnFCUytl0WolZmlFWGxYTG00cElCVDFRbGFJQlJRMDRyQ0JZbHdCNlVFbV
B
lWjhuQ31mR2JYTC9HM2wKcXBYTDVKdzJqcVh2MXdzcWsrdWNCRkOzZOFYZkSYZkhlRExnVOVyNXRZRlF4VXo5UFF
HODl
pcDY10TBkYnBlSApOMnU2NGk4UTgldk83OFVITlc2eUFZUlloZVdha093RDFwZzNPdkhxV3FhbnVlMnlrOHxaUUR0WW5
2MytBeU5DCnloNlRaRHluZ01ZdEptbDFTQ01TNF.pSR2d4NXNwaCtK0C9X0Gx0Ri9wMWZxbTA0bXZSRndxU3M2YlJCQ2
Z
PWcKbFVlMGxLRUNnWUVBNWJzT01WzFBVndJTltkTscOpSVDNURkI2OVlxbDRYcnZRR0FZY3BhdktENnd5VmtEOTVl
QQp
SaXVRSlNNKzY4REtBVmlpYllpaThJemExTkdqdC9JZDUwTGVoNklaRVg2enVpK0g3dlBSbVd6SE9ueWNmU2FmClVQME
F
RL0RiM21CNWJQTmJHYXNkaDNIb2JvR0NSSHZmfrFFXY2tYbUVXM2ZudVHRlJLZ2xlTEVDZllFQTVUdysKTEVTVlBE
SFF
mamNBN0htNmdsMndGRidCUGlsSGdaYWRN25F.b3ZvRmMxalBMRHVCMWJ6OHJNWldleGdmaHN0OQpMZ0xSUDBXdk
JWdlJ
sVrdMTmFbTlVzRnikxU2dvaWZsS01ZWkMyZmpLWTYlRFE3YUUzcTdnVis4U2pIZHpoclhCCkVQclAvWXQ3S0QrbFBMZ
mh
aNXNKZWFtelY3b3gveno5Y0s0U0ZKc0NnWUJ4OVk2VzFydHBoMFcvS05JS3V4SEoKMjRxRFQxbmlObE9FdmFhakFUarj
ZQkxXYnIvWERsNWRjTEs4bFcxYkNYR3JwY2s3U0xKNlhZUVlXajQ2dTNJMgpEQ2ZUWlFiRWRQTzNBbWtPR2ZqWmdP
cDd
pdaVkL0JDLzNpRkprcXVlenNFdFdMrHlVcjM5T0hZeW16QVJ4Tmo2CnZuUGlma000Rkl6d3F2MHVoN0xlblFLQmdBTlZ
XZTRZMlRwbzJ3aEswbmVkMllsMXhVNjJoZ2JiVHcvaVdhdVcKY3ZMV3dlZUlmd0Q4MVRWL2R3a29KVEMlVEJRUXQZ
Ukk
xRFFnVmtnMHFWcUtOOGhDNGQuH05MRzIwTWdZMk94WgpjSFZzK2J4elYwWB6VlRUbEMyVEsyamhmOHVRcndzSktx
Y2N
```

www.atmicnetworks.com

www.atmicnetworks.com

```
OU0ZEczZwclhocThsV213Znd3aWlBRlhLSFJRCkE3RkxBb0dCQUx3NW8rbHFVZ3hHQlpKdy9EelRGekSTekQreVd6Um8
lc2ZEc2x6a2FvY0pUbEx2MUNndEVIc3QKeG5HMTlIYStSMlM3cDRtei9beDJYMFRzaTZzUzVwWlR5WEx5STF5azh2TUZ
rRldacjRmeVtlXV2t3SjZlVEHYwpyWF13TWM5VFlDUGZrSFJaTm9XRlhZV3BkeTJBOXZCbFlScHZsQVZoenU2TlVZQ2w
5b2ZpCi0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
rootSkschOOlOl-master:~i vim /etc/kubernetes/manitests/kube-apiserver.yaml I
```

www.atmicnetworks.com

```
root@ksch00101-master:~l vim /etc/kube metes/man! fests/kube-apiserver .yaml
root@kschOOlO1—master:~# systemctl daemon-reload
srootSkschOO101 master:-4 systemctl restart kubelet.service rootSkschOOl01 master:-! kubectl
get nodes
error: You must be logged in to the server (Unauthorized) rootSkschOOiOl-master:~i exit logout
Connection to 10.240.86.190 closed.
candidate@cli :~$ kubectl get nodes
KAME                    STATUS ROLES                    AGE VERSION


kschOO10!-master        Ready      control-plane,master      93d yl.    .23.3
kschOOlOl-workerl       Ready      <none>                    93d vl.    .23.3
candidate@cli:~$ kubectl get pod -n kube-system
KAME                                      READ    STATUS    RESTARTS        AGE
coredns-64897985d-7pnhm                   1/1     Running   1 (7h2m ago)    93d
coredns—64897985d—rrVsd                   1/1     Running   1 (7h2m ago)    93d
etcd-kschOOl01-master                     1/1     Running   1 (7h2m ago)    93d
kube aplserver-kschOOlOl-master           0/1     Running   0               24s
kube-controller-manager-kschQOl01-master  1/1     Running   3 (42s ago)     93d
kube-ilannel-ds-llktn                     1/1     Running   1 (93d ago)     93d
kube-flanne1 -ds-q9ynl                    1/1     Running   1 (93d ago)     93d
kube-proxy-2c4ht                          1/1     Running   1 (93d ago)     93d
kube-proxy-pmmbc                          1/1     Running   1 (93d ago)     93d
kube-scheduler-kschOOl01-master candidate@cli:~$  1/1     Running   3 (42s ago)     93d
kubectl get pod -n kube-                  system
NAME                                      READ    STATUS    RESTARTS        AGE
coredns-64897985d-7pnhm                   1/1     Running   1 (7h2m ago)    93d
coredns-64897985d-rr7 sd                  1/1     Running   1 (7h2m ago)    93d
etcd-kschOOlOl-master                     1/1     Running   1 (7h2m ago)    93d
kube apiserver kschOOlOl master           0/1     Running   0               30s
kube-controller-manager-kschOOlOl-master  1/1     Running   3 (48s ago)     93d
kube-flanne1-ds-llktn                     1/1     Running   1 (93d ago)     93d
kube-f1annel-ds-q9vnl                     1/1     Running   1 (93d ago)     93d
kube-proxy-2c4ht                          1/1     Running   1 (93d ago)     93d
kube-proxy-pmmbc                          1/1     Running   1 (93d ago)     93d
kube-scheduler-kschOOlOl-master           1/1     Running   3 (48s ago)     93d
candidate@cli:~$ kubectl get clusterrolebindings,                rbac,authorization.k8s. io   1 grep anon
system:anonymous                                                 ciusterRole/cluster-         admin
                                                  7him
candidate@cli:~? kubectl delete clusterroiebindings.rbac.authorization.kBs.io/system:anonymo us
clusterrolebinding.rbac,authorization.kBs.io "system:anonymous" deleted
```

Question: 49

SIMULATION

You must connect to the correct host. Failure to do so may result In a zero score,

ssh ck5000002

Context

www.atmicnetworks.com

You must resolve issues that a CIS Benchmark tool found for the kubeadm provisioned cluster.

Task

Fix all issues via configuration and restart the affected components to ensure the new settings take effect.

Fix all of the following violations that were found against the kubelet:

The cluster uses the Docker Engine os its container runtime, If needed, use the

docker command to troubleshaot running containers.

Ensure that the anonymous-auth argument is set to false FAIL

Ensure that the -authorization-mode argument is not set to FAIL

AlwaysAllow

Use Webhook authentication /authorization where possible.

Fix all of the following violations that were found against ettd :

Ensure that the -client cert auth argument is set to true FAIL

www.atmicnetworks.com

Answer: See the Explanation below for complete solution.

Explanation:

1) SSH to the right node

ssh cks000002

sudo -i

2) Fix kubelet CIS findings

2.1 Edit kubelet config (MAIN place in kubeadm clusters)

vi /var/lib/kubelet/config.yaml

A) Set anonymous-auth to false

Find (or add) this block exactly:

authentication:

anonymous:

enabled: false

B) Use Webhook authentication (recommended by task)

Ensure this exists under authentication:

webhook:

enabled: true

C) Use Webhook authorization and NOT AlwaysAllow

Find (or add) this block exactly:

authorization:

mode: Webhook

When done, your file should contain something like this (exact structure to aim for):

www.atmicnetworks.com

authentication:

anonymous:

enabled: false

webhook:

enabled: true

x509:

clientCAFile: /etc/kubernetes/pki/ca.crt

authorization:

mode: Webhook

If x509: section isn't there, it's usually already present in kubeadm; don't panic. Only the task- required parts are: anonymous false + webhook enabled + authorization mode Webhook.

2.2 Restart kubelet (required for config.yaml changes)

systemctl daemon-reload

systemctl restart kubelet

systemctl status kubelet --no-pager

Quick confirm (optional but fast):

grep -nE "anonymous|webhook|authorization|mode" /var/lib/kubelet/config.yaml

3) Fix etcd CIS finding: --client-cert-auth=true

3.1 Edit etcd static pod manifest (kubeadm path)

vi /etc/kubernetes/manifests/etcd.yaml

Find the container command: args that look like:

- command:

- etcd

- --something=...

Ensure this line exists exactly in the list:

- --client-cert-auth=true

Also ensure this is present (usually already is, but add if missing):

- --trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt

Example snippet (what you want the args area to include):

- command:

- etcd

- --client-cert-auth=true

- --trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt

3.2 Apply etcd change (auto-restart happens)

Just save the file. Kubelet will restart etcd automatically.

Watch it restart (pick one depending on runtime):

If Docker runtime (your task mentions Docker):

docker ps | grep etcd

If you don't see it briefly, wait 2–5 seconds and rerun:

docker ps | grep etcd

(Alternative if available)

crictl ps | grep etcd

4) Final quick validation (fast exam check)

Kubelet config check

grep -n "enabled: false" -n /var/lib/kubelet/config.yaml | head

grep -n "webhook" /var/lib/kubelet/config.yaml

grep -n "authorization" /var/lib/kubelet/config.yaml

etcd arg check

grep -n "client-cert-auth" /etc/kubernetes/manifests/etcd.yaml

www.atmicnetworks.com

Question: 50

SIMULATION

Context

For testing purposes, the kubeadm provisioned cluster 's API server was configured to allow unauthenticated and unauthorized access.

Task

First, secure the cluster 's API server configuring it as follows:

. Forbid anonymous authentication

. Use authorization mode Node,RBAC

. Use admission controller NodeRestriction

The cluster uses the Docker Engine as its container runtime . If needed, use the docker command to troubleshoot running containers.

kubectl is configured to use unauthenticated and unauthorized access. You do not have to change it, but be aware that kubectl will stop working once you have secured the cluster .

You can use the cluster 's original kubectl configuration file located at etc/kubernetes/admin.conf to access the secured cluster.

Next, to clean up, remove the ClusterRoleBinding

system:anonymous.

Answer: See the Explanation below for complete solution.

Explanation:

1) SSH to control-plane node

ssh cks000002

sudo -i

2) Edit API Server static pod manifest

API server in kubeadm runs as a static pod.

vi /etc/kubernetes/manifests/kube-apiserver.yaml

3) Apply required API Server security settings

3.1 Forbid anonymous authentication

Find command: section and ensure this line exists:

- --anonymous-auth=false

3.2 Use authorization mode Node,RBAC

Ensure exactly this line exists (and no AlwaysAllow):

- --authorization-mode=Node,RBAC

X Remove if present:

- --authorization-mode=AlwaysAllow

www.atmicnetworks.com

3.3 Enable admission controller NodeRestriction

Find --enable-admission-plugins and ensure NodeRestriction is included.

Correct example:

- --enable-admission-plugins=NodeRestriction

If other plugins already exist, append NodeRestriction, e.g.:

- --enable-admission-plugins=NamespaceLifecycle,ServiceAccount,NodeRestriction

4) Save file and let kubelet restart API server

Just save and exit (:wq)

Kubelet will automatically restart the API server pod.

5) Switch kubectl to secured config

! Current kubectl will stop working after API server hardening.

export KUBECONFIG=/etc/kubernetes/admin.conf

Verify access:

kubectl get nodes

6) Remove insecure ClusterRoleBinding

Delete system:anonymous binding:

kubectl delete clusterrolebinding system:anonymous

Verify removal:

kubectl get clusterrolebinding | grep anonymous

(no output = correct)

7) Quick validation (optional but fast)

API server flags check:

grep -n "anonymous-auth" /etc/kubernetes/manifests/kube-apiserver.yaml

grep -n "authorization-mode" /etc/kubernetes/manifests/kube-apiserver.yaml

www.atmicnetworks.com

grep -n "NodeRestriction" /etc/kubernetes/manifests/kube-apiserver.yaml

Question: 51

SIMULATION

Context

You must fully integrate a container image scanner into the kubeadm provisioned cluster.

Task

Given an incomplete configuration located at /etc/kubernetes/bouncer and a functional container image scanner

with an HTTPS endpoint at https://smooth-yak.local/review, perform the following tasks to implement a validating admission controller.

First, re-configure the API server to enable all admission plugin(s) to support the provided AdmissionConfiguration.

Next, re-configure the ImagePolicyWebhook configuration to deny images on backend failure.

Next, complete the backend configuration to point to the container image scanner's endpoint at https://smooth-yak.local/review.

Finally, to test the configuration, deploy the test resource defined in /home/candidate/vulnerable.yaml which is using an image that should be denied.

You may delete and re-create the resource as often as needed.

The container image scanner's log file is located at /var/log/nginx/access_log.

www.atmicnetworks.com

Answer: See the Explanation below for complete solution.

Explanation:

Below is the CKS exam style “do-this-exactly” runbook for Q3. It includes the minimal discovery commands (so you don’t guess filenames), then the exact lines/blocks to set.

QUESTION 3 – ImagePolicyWebhook (Validating Admission) – Exam Steps

0) SSH + root

ssh cks000002

sudo -i

1) Identify the provided config files (no guessing)

ls -la /etc/kubernetes/bouncer

You are looking for files typically named like:

admission_configuration.yaml (AdmissionConfiguration)

imagepolicywebhook.yaml (ImagePolicyWebhookConfiguration) OR the ImagePolicyWebhook config embedded inside the AdmissionConfiguration

kubeconfig (webhook kubeconfig)

If unsure which is which, quick peek:

grep -R "ImagePolicyWebhook" -n /etc/kubernetes/bouncer

grep -R "AdmissionConfiguration" -n /etc/kubernetes/bouncer

grep -R "kubeconfig" -n /etc/kubernetes/bouncer

PART A — Reconfigure API Server to enable required admission plugin(s)

2) Edit API server static pod manifest

vi /etc/kubernetes/manifests/kube-apiserver.yaml

2.1 Enable the admission plugin ImagePolicyWebhook

Find the line starting with:

- --enable-admission-plugins=

Ensure ImagePolicyWebhook is included in that comma list.

Example (your list may differ; just add ImagePolicyWebhook):

- --enable-admission-plugins=NodeRestriction,ImagePolicyWebhook

If the flag does not exist, add one line under command::

- --enable-admission-plugins=ImagePolicyWebhook

2.2 Point API server to the provided AdmissionConfiguration

In the same file, ensure this flag exists (use the file in /etc/kubernetes/bouncer that contains AdmissionConfiguration):

- --admission-control-config-file=/etc/kubernetes/bouncer/admission_configuration.yaml

If your file is named differently, use the real filename you found in step 1, but keep the flag name exactly --admission-control-config-file.

Save/exit:

:wq

Static pod will restart automatically (kubelet watches the manifest).

Optional quick watch:

docker ps | grep kube-apiserver

or:

crictl ps | grep kube-apiserver

PART B — Configure ImagePolicyWebhook to deny images on backend failure

3) Edit the ImagePolicyWebhook config

One of these is true on your cluster:

Option 1 (most common in these tasks): ImagePolicyWebhook config is a standalone file

Edit the file in /etc/kubernetes/bouncer that contains kind: ImagePolicyWebhookConfiguration:

grep -R "kind: ImagePolicyWebhookConfiguration" -n /etc/kubernetes/bouncer

vi /etc/kubernetes/bouncer/<THE_FILE_YOU_FOUND>.yaml

Set (or ensure) exactly:

defaultAllow: false

www.atmicnetworks.com

Option 2: ImagePolicyWebhook config is embedded inside AdmissionConfiguration

Edit the AdmissionConfiguration file:

vi /etc/kubernetes/bouncer/admission_configuration.yaml

Find the plugin section for ImagePolicyWebhook and ensure the config includes:

defaultAllow: false

Q Save/exit:

:wq

PART C — Point backend configuration to https://smooth-yak.local/review

4) Edit the webhook kubeconfig to use the scanner endpoint

Find the kubeconfig file referenced by the ImagePolicyWebhook config.

Search for kubeConfigFile:

grep -R "kubeConfigFile" -n /etc/kubernetes/bouncer

Open that kubeconfig path (example name below; yours may differ):

vi /etc/kubernetes/bouncer/kubeconfig

In kubeconfig, set the cluster server exactly:

clusters:

- cluster:

server: https://smooth-yak.local/review

Q Save/exit:

:wq

PART D — Restart effect (make sure API server picks up config)

Because you already edited /etc/kubernetes/manifests/kube-apiserver.yaml, the API server restarted.

To be safe (and fast), force a restart by “touching” the manifest (no content change needed):

touch /etc/kubernetes/manifests/kube-apiserver.yaml

www.atmicnetworks.com

PART E — Test: apply vulnerable workload and confirm it is denied

5) Use admin kubeconfig (because old kubectl config may break)

export KUBECONFIG=/etc/kubernetes/admin.conf

kubectl get nodes

6) Deploy the test resource (should be DENIED)

kubectl apply -f /home/candidate/vulnerable.yaml

Expected: admission error/denied message.

If it already exists:

kubectl delete -f /home/candidate/vulnerable.yaml

kubectl apply -f /home/candidate/vulnerable.yaml

PART F — Verify the scanner was called (log check)

7) Check scanner access log

tail -n 50 /var/log/nginx/access_log

You should see requests hitting /review.

Quick “what to check if it doesn’t deny”

Run these in order:

Confirm API server flags:

grep -n "enable-admission-plugins" /etc/kubernetes/manifests/kube-apiserver.yaml

grep -n "admission-control-config-file" /etc/kubernetes/manifests/kube-apiserver.yaml

Confirm deny-on-failure:

grep -R "defaultAllow" -n /etc/kubernetes/bouncer

Must show:

defaultAllow: false

Confirm endpoint:

grep -R "server: https://smooth-yak.local/review" -n /etc/kubernetes/bouncer

API server logs (docker runtime):

docker ps | grep kube-apiserver

docker logs $(docker ps -q --filter name=kube-apiserver) --tail 80

If you paste the output of:

ls -la /etc/kubernetes/bouncer

grep -R "kind: AdmissionConfiguration" -n /etc/kubernetes/bouncer

grep -R "ImagePolicyWebhook" -n /etc/kubernetes/bouncer

Question: 52

SIMULATION

You must connect to the correct host . Failure to do so may result in a zero score.

[candidato@base] $ ssh cks000023

Task

Analyze and edit the Dockerfile located at /home/candidate/subtle-bee/build/Dockerfile, fixing one instruction present in the file that is a prominent security/best-practice issue.

Do not add or remove instructions; only modify the one existing instruction with a security/best- practice concern.

Do not build the Dockerfile, Failure to do so may result in running out of storage and a zero score.

Analyze and edit the given manifest file /home/candidate/subtle-bee/deployment.yaml, fixing one fields present in the file that are a prominent security/best-practice issue.

Do not add or remove fields; only modify the one existing field with a security/best-practice concern.

www.atmicnetworks.com

Should you need an unprivileged user for any of the tasks, use user nobody with user ID 65535.

Answer: See the Explanation below for complete solution.

Explanation:

0) Connect to the correct host

ssh cks000023

sudo -i

PART A — Fix ONE prominent Dockerfile security/best-practice issue

1) Open the Dockerfile

vi /home/candidate/subtle-bee/build/Dockerfile

2) Find the “most obvious” security/best-practice problem and modify ONLY THAT ONE instruction

Use / search in vi to quickly find candidates:

Candidate 1 (very common): USER root (or no USER but a USER 0)

Search:

/USER

If you see:

USER root

Change that single instruction to:

USER 65535

(or USER nobody if that exact word is already used in the file—but the task explicitly allows UID 65535, so USER 65535 is safest.)

Q This is one-instruction change and is a top-tier best practice.

Candidate 2 (very common): FROM <image>:latest

Search:

www.atmicnetworks.com

/FROM

If you see something like:

FROM nginx:latest

Change ONLY that line to a pinned tag (example):

FROM nginx:1.25.5

(Any non-latest pinned version is the point. Don't add a digest line; just modify the existing FROM line.)

Candidate 3: ADD http://... (remote URL download)

Search:

/ADD

If you see remote URL usage like:

ADD https://example.com/app.tar.gz /app/

Change that single instruction to COPY only if it's copying local files.

If it's a remote URL, the more "correct" fix would normally be using curl with verification, but that would require adding instructions (not allowed).

So in this exam constraint, do NOT pick this unless it's actually a local add like:

ADD . /app

Then change just the word:

COPY . /app

3) Save and exit

:wq

! Don't run docker build (task forbids building).

PART B — Fix ONE prominent security/best-practice issue in the Deployment manifest

4) Open the manifest

vi /home/candidate/subtle-bee/deployment.yaml

5) Change ONLY ONE existing field that is a clear security issue

Use / search in vi for the usual “bad fields”:

Option 1 (most common): running as root

Search:

/runAsUser

If you see:

runAsUser: 0

Change that one existing field value to:

runAsUser: 65535

Q This is a single-field change and matches the prompt hint.

Option 2: privileged container

Search:

/privileged

If you see:

privileged: true

Change only that value to:

privileged: false

Option 3: allow privilege escalation

Search:

/allowPrivilegeEscalation

If you see:

allowPrivilegeEscalation: true

Change only that value to:

allowPrivilegeEscalation: false

Option 4: writable root filesystem

www.atmicnetworks.com

www.atmicnetworks.com

Search:

/readOnlyRootFilesystem

If you see:

readOnlyRootFilesystem: false

Change only that value to:

readOnlyRootFilesystem: true

Option 5: image uses :latest

Search:

/image:

If you see:

image: something:latest

Change only that value to a pinned tag, e.g.:

image: something:1.2.3

6) Save and exit

:wq

What to pick (fast decision rule)

If you see run as root in either file, that's usually the highest scoring / most "prominent" security issue.

Dockerfile: USER root → USER 65535

Deployment: runAsUser: 0 → runAsUser: 65535

Those are perfect because you only modify one line/field and it matches the hint.

www.atmicnetworks.com

Question: 53

SIMULATION

Documentation Deployments, Pods, Falco

You must connect to the correct host . Failure to do so may result in a zero score.

[candidate@base] $ ssh cks000026

Context

A Pod is misbehaving and poses a security threat to the system.

Task

One of the Pods belonging to the application ollama is misbehaving. It is directly accessing the system's memory reading from the sensitive file /dev/mem.

First, identify the misbehaving Pod accessing /dev/mem.

The cluster uses the Docker Engine as its container runtime . If needed, use the docker command to troubleshoot running containers.

Next, identify the Deployment managing the misbehaving Pod and scale it to zero replicas.

Do not modify the Deployment except for scaling it down.

Do not modify any other Deployments .

Do not delete any Deployments.

www.atmicnetworks.com

Answer: See the Explanation below for complete solution.

Explanation:

1) Connect to the correct host

ssh cks000026

sudo -i

2) Identify the misbehaving Pod accessing /dev/mem
This task hints Falco is available → use it first (fast + intended).

2.1 Check Falco logs for /dev/mem access

journalctl -u falco | grep dev/mem

If Falco runs as a pod instead of systemd:

kubectl -n falco logs -l app=falco | grep dev/mem

2.2 Identify the Pod name

From the Falco output, you will see something like:

Pod=ollama-xxxxx Namespace=default File=/dev/mem

Note the exact Pod name (example: ollama-7c9d6f7b6d-abcde)

3) (If Falco logs are unclear) Confirm using Docker runtime

Because the cluster uses Docker, verify which container is accessing /dev/mem.

3.1 List running containers

docker ps

3.2 Inspect suspicious container

(Find container related to ollama)

docker inspect <container_id> | grep ollama

www.atmicnetworks.com

You should confirm it maps to the same Pod you saw in Falco.

4) Identify the Deployment managing the misbehaving Pod

4.1 Get Pod details

kubectl get pod <MISBEHAVING_POD_NAME> -o wide

4.2 Find owning Deployment

kubectl get pod <MISBEHAVING_POD_NAME> -o jsonpath='{.metadata.ownerReference[0].name}'

This will output something like:

ollama

That is the Deployment name

5) Scale ONLY that Deployment to zero replicas

! Do not edit, delete, or touch anything else

kubectl scale deployment ollama --replicas=0

6) Verify the Pod is terminated kubectl get pods | grep ollama Expected: no running Pods Also confirm replicas: kubectl get deployment ollama Replicas should show: 0/0

Question: 54

SIMULATION

Documentation Deployment, Pod, Namespace

You must connect to the correct host . Failure to do so may result in a zero score.

[candidate@base] $ ssh cks000028

Context

www.atmicnetworks.com

You must update an existing Pod to ensure the immutability of its containers.

Task

Modify the existing Deployment named lamp-deployment, running in namespace lamp, so that its containers:

. run with user ID 20000

. use a read-only root filesystem

. forbid privilege escalation

The Deployment's manifest file con be found at /home/candidate/finer-sunbeam/lamp-deployment.yaml.

Answer: See the Explanation below for complete solution.

Explanation:

1) Connect to the correct host

ssh cks000028

sudo -i

2) Use the right kubeconfig (safe in exam)

export KUBECONFIG=/etc/kubernetes/admin.conf

3) Open the provided Deployment manifest

vi /home/candidate/finer-sunbeam/lamp-deployment.yaml

4) Edit ONLY the Pod template security settings (add/modify these fields)

Inside:

spec: -> template: -> spec:

4.1 Set container to run as user 20000

Add (or change) under the container securityContext::

www.atmicnetworks.com

securityContext:

runAsUser: 20000

4.2 Make root filesystem read-only

In the SAME container securityContext: ensure:

readOnlyRootFilesystem: true

4.3 Forbid privilege escalation

In the SAME container securityContext: ensure:

allowPrivilegeEscalation: false

Q The container section should look like this (example — keep your existing image/ports/etc):

```
spec:
  template:
    spec:
      containers:
      - name: <your-container-name>
        image: <unchanged>
        securityContext:
          runAsUser: 20000
          readOnlyRootFilesystem: true
          allowPrivilegeEscalation: false
```

If there are multiple containers, apply the same securityContext to each container.

Save and exit:

:wq

5) Apply the manifest (updates Deployment -> recreates Pods)

kubectl -n lamp apply -f /home/candidate/finer-sunbeam/lamp-deployment.yaml

6) Wait for rollout

www.atmicnetworks.com

kubectl -n lamp rollout status deployment/lamp-deployment

7) Verify the security settings are live

7.1 Check the Pod is running

kubectl -n lamp get pods -l app=lamp -o wide

(if label differs, just kubectl -n lamp get pods)

7.2 Verify the three fields on a running Pod

Pick the Pod name and run:

POD=$(kubectl -n lamp get pods -o jsonpath='{.items[0].metadata.name}')

kubectl -n lamp get pod $POD -o jsonpath='{.spec.containers[0].securityContext.runAsUser}{"\n"}{.spec.containers[0].securityContext.readOnlyRootFilesystem}{"\n"}{.spec.containers[0].securityContext.allowPrivilegeEscalation}{"\n"}'

Expected output:

20000

true

false

If the pod fails after readOnlyRootFilesystem=true

Don't change the requirement (task demands it). Usually the app needs writable dirs via volumes, but the task doesn't ask for that—so only adjust if the manifest already has volumes and just needs these securityContext fields.

Question: 55

SIMULATION

Context

You must implement auditing for the kubeadm provisioned cluster.

Task

First, reconfigure the cluster 's API server, so that:

. the basic audit policy located at

/etc/kubernetes/logpolicy/audit-policy.yaml is used,

. logs are stored at /var/log/kubernetes/audit-logs.txt,

. and a maximum of 2 logs are retained for 10 days.

The cluster uses the Docker Engine as its container runtime . If needed, use the docker command to troubleshoot running containers.

The basic policy only specifies what not to log.

Next, edit and extend the basic policy to log:

. namespaces interactions at RequestResponse level

. the request body of deployments interactions in the namespace webapps

. ConfigMap and Secret interactions in all namespaces at the Metadata level

. all other requests at the Metadata level

Make sure the API server uses the extended policy.

Failure to do so may result in a reduced score.

Answer: See the Explanation below for complete solution.

Explanation:

1) Connect to the correct host

ssh cks000028

sudo -i

www.atmicnetworks.com

(If hostname differs in your exam, use the one shown in the question banner.)

2) Edit the API server static pod manifest

API server is a static pod in kubeadm.

vi /etc/kubernetes/manifests/kube-apiserver.yaml

3) Configure API server to enable auditing

Inside the command: section, ensure ALL of the following flags exist (add them if missing, modify if present).

3.1 Use the given audit policy file

- --audit-policy-file=/etc/kubernetes/logpolicy/audit-policy.yaml

3.2 Store audit logs at the required location

- --audit-log-path=/var/log/kubernetes/audit-logs.txt

3.3 Retain a maximum of 2 log files

- --audit-log-maxbackup=2

3.4 Retain logs for 10 days

- --audit-log-maxage=10

Q Example (your file may have more flags — that's fine):

- command:
- kube-apiserver
- --audit-policy-file=/etc/kubernetes/logpolicy/audit-policy.yaml
- --audit-log-path=/var/log/kubernetes/audit-logs.txt
- --audit-log-maxbackup=2
- --audit-log-maxage=10

Save and exit:

:wq

The API server will auto-restart (static pod).

www.atmicnetworks.com

Optional quick check:

```
docker ps | grep kube-apiserver
```

4) Edit and EXTEND the audit policy

Open the given basic policy:

```
vi /etc/kubernetes/logpolicy/audit-policy.yaml
```

The file already contains rules for what NOT to log.
You must ADD rules BELOW them (do not delete existing ones).

5) Add the required audit rules (EXACT OtRDER)

Append the following rules in this order (order matters in audit policies).

5.1 Log namespaces interactions at RequestResponse

```
-level: RequestResponse
resources:
- group: ""
  resources: ["namespaces"]
```

5.2 Log deployment request bodies in namespace webapps

```
- level: RequestResponse
namespaces: ["webapps"]
resources:
- group: "apps"
  resources: ["deployments"]
```

5.3 Log ConfigMap and Secret interactions (all namespaces) at Metadata

www.atmicnetworks.com

-level: Metadata

resources:

- group: ""

resources: ["configmaps", "secrets"]

5.4 Log all other requests at Metadata

! This must be LAST

- level: Metadata

5.5 Final audit-policy.yaml should END like this

- (existing "do not log" rules above)

- level: RequestResponse

resources:

- group: ""

resources: ["namespaces"]

- level: RequestResponse

namespaces: ["webapps"]

resources:

- group: "apps"

resources: ["deployments"]

- level: Metadata

resources:

- group: ""

www.atmicnetworks.com

resources: ["configmaps", "secrets"]

- level: Metadata

Save and exit:

:wq

6) Make sure API server uses the EXTENDED policy

Touch the manifest to guarantee reload:

touch /etc/kubernetes/manifests/kube-apiserver.yaml

Wait a few seconds.

7) Verify auditing is working

7.1 Check audit log file exists

ls -l /var/log/kubernetes/audit-logs.txt

7.2 Generate test activity

kubectl get namespaces

kubectl get configmaps -A

7.3 Confirm logs are written

tail -n 20 /var/log/kubernetes/audit-logs.txt

You should see audit entries.

Question: 56

SIMULATION

Documentation Namespace, NetworkPolicy, Pod

You must connect to the correct host . Failure to do so may result in a zero score.

www.atmicnetworks.com

[candidate@base] $ ssh cks000031

Context

You must implement NetworkPolicies controlling the traffic flow of existing Deployments across namespaces.

Task

First, create a NetworkPolicy named deny-policy in the prod namespace to block all ingress traffic.

The prod namespace is labeled env:prod

Next, create a NetworkPolicy named allow-from-prod in the data namespace to allow ingress traffic only from Pods in the prod namespace.

Use the label of the prod names & Click to copy traffic.

The data namespace is labeled env:data

Do not modify or delete any namespaces or Pods . Only create the required NetworkPolicies.

Answer: See the Explanation below for complete solution.

Explanation:

1) Connect to the correct host

ssh cks000031

sudo -i

2) Use admin kubeconfig (safe default)

export KUBECONFIG=/etc/kubernetes/admin.conf

www.atmicnetworks.com

PART A — Deny ALL ingress traffic in prod namespace

Requirement:

NetworkPolicy name: deny-policy

Namespace: prod (namespace is labeled env=prod)

Effect: block all ingress

3) Create deny-policy in prod

Create the policy directly with kubectl (fastest & safest):

```
cat <<EOF | kubectl apply -f -
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: deny-policy
namespace: prod
spec:
podSelector: {}
policyTypes:
- Ingress
EOF
```

Q What this does:

podSelector: {} → selects all Pods in prod

No ingress: rules → deny all ingress traffic

4) Verify

kubectl -n prod get networkpolicy deny-policy

PART B — Allow ingress to data ONLY from Pods in prod

Requirement:

www.atmicnetworks.com

NetworkPolicy name: allow-from-prod

Namespace: data (namespace is labeled env=data) Allow ingress only from Pods in prod namespace

Use namespace label (env=prod)

5) Create allow-from-prod policy in data

```
cat <<EOF | kubectl apply -f -
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-from-prod
namespace: data
spec:
podSelector: {}
policyTypes:
- Ingress
ingress:
- from:
  - namespaceSelector:
      matchLabels:
        env: prod
EOF
```

Q What this does:

Applies to all Pods in data

Allows ingress only from namespaces labeled env=prod

All other ingress traffic is denied by default

6) Verify

kubectl -n data get networkpolicy allow-from-prod

FINAL CHECK (What the examiner expects)

kubectl get networkpolicy -n prod

kubectl get networkpolicy -n data

You should see:

deny-policy in prod

allow-from-prod in data

Question: 57

SIMULATION

Documentation Ingress, Service, NGINX Ingress Controller

You must connect to the correct host . Failure to do so may result in a zero score.

[candidate@base] $ ssh cks000032

Context

You must expose a web application using HTTPS routes.

Task

Create an Ingress resource named web in the prod namespace and configure it as follows:

. Route traffic for host web.k8s.local and all paths to the existing Service web

. Enable TLS termination using the existing Secret web-cert.

. Redirect HTTP requests to HTTPS .

You can test your Ingress configuration with the following command:

www.atmicnetworks.com

[candidate@cks000032]$ curl -L http://web.k8s.local

Answer: See the Explanation below for complete solution.

Explanation:

1) Connect to the correct host

ssh cks000032

sudo -i

2) Use admin kubeconfig

export KUBECONFIG=/etc/kubernetes/admin.conf

3) Verify prerequisites (quick check)

These should already exist per task.

kubectl -n prod get svc web

kubectl -n prod get secret web-cert

kubectl get pods -n ingress-nginx

(If the ingress controller pods exist, you're good.)

4) Create the Ingress resource

Create Ingress named web in namespace prod with:

host: web.k8s.local

all paths → Service web

TLS using Secret web-cert

HTTP → HTTPS redirect (NGINX)

cat <<EOF | kubectl apply -f -

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

www.atmicnetworks.com

```
name: web
namespace: prod
annotations:
  nginx.ingress.kubernetes.io/ssl-redirect: "true"
  nginx.ingress.kubernetes.io/force-ssl-redirect: "true"
spec:
ingressClassName: nginx
tls:
- hosts:
  - web.k8s.local
secretName: web-cert
rules:
- host: web.k8s.local
  http:
    paths:
    - path: /
      pathType: Prefix
      backend:
        service:
          name: web
          port:
            number: 80
EOF
```

5) Verify Ingress creation

```
kubectl -n prod get ingress web kubectl -n prod describe ingress web
```

www.atmicnetworks.com

Confirm:

Host = web.k8s.local

TLS Secret = web-cert

Backend Service = web

6) Test HTTP → HTTPS redirect

curl -L http://web.k8s.local

Expected:

Redirects to https://web.k8s.local

Returns application response over HTTPS

Question: 58

SIMULATION

Documentation

ServiceAccount, Deployment,

Projected Volumes

You must connect to the correct host . Failure to do so may

result in a zero score.

[candidate@base] $ ssh cks000033

Context

A security audit has identified a Deployment improperly handling service account tokens, which could lead to security vulnerabilities.

Task

First, modify the existing ServiceAccount stats-monitor-sa in the namespace monitoring to turn off automounting of API credentials.

Next, modify the existing Deployment stats-monitor in the namespace monitoring to inject a ServiceAccount token mounted at /var/run/secrets/kubernetes.io/serviceaccount/token.

Use a Projected Volume named token to inject the ServiceAccount token and ensure that it is mounted read-only.

The Deployment's manifest file can be found at /home/candidate/stats-monitor/deployment.yaml.

Answer: See the Explanation below for complete solution.

Explanation:

1) Connect to correct host

ssh cks000033

sudo -i

export KUBECONFIG=/etc/kubernetes/admin.conf

2) Patch the ServiceAccount to disable automounting

Task: turn off automounting of API credentials for stats-monitor-sa in monitoring.

kubectl -n monitoring patch sa stats-monitor-sa -p '{"automountServiceAccountToken": false}'

Verify:

kubectl -n monitoring get sa stats-monitor-sa -o yaml | grep -i automount

3) Edit the Deployment manifest file

Task says to modify the manifest at:

/home/candidate/stats-monitor/deployment.yaml

www.atmicnetworks.com

vi /home/candidate/stats-monitor/deployment.yaml

4) In the Deployment, ensure it uses the ServiceAccount AND inject token via Projected Volume

4.1 Make sure Deployment uses the SA

Under:

spec: -> template: -> spec:

ensure:

serviceAccountName: stats-monitor-sa

(If it already exists, leave it; don't add extra changes beyond requirements.)

4.2 Add a projected volume named token

Under:

spec: -> template: -> spec: -> volumes:

add (or modify existing volume if present) so it is exactly:

- name: token

projected:

sources:

- serviceAccountToken:

path: token

This creates the file token inside the mounted directory, so the final path becomes:

/var/run/secrets/kubernetes.io/serviceaccount/token

4.3 Mount the projected volume read-only at the required location

Under the target container:

spec: -> template: -> spec: -> containers: -> (your container) -> volumeMounts:

Add:

- name: token

mountPath: /var/run/secrets/kubernetes.io/serviceaccount

readOnly: true

www.atmicnetworks.com

Q This satisfies:

Projected volume name: token

Mount path: /var/run/secrets/kubernetes.io/serviceaccount/token (file inside mount)
Mounted read-only

4.4 Important: Don't break default token mount behavior

Because you disabled SA automounting at the ServiceAccount level, you must explicitly mount the projected token (done above). That's the whole point of this task.

Save and exit:

:wq

5) Apply the updated Deployment

kubectl -n monitoring apply -f /home/candidate/stats-monitor/deployment.yaml

Wait rollout:

kubectl -n monitoring rollout status deployment/stats-monitor

6) Verify the token file exists in the running Pod

Get a pod name:

POD=$(kubectl -n monitoring get pods -l app=stats-monitor -o jsonpath='{.items[0].metadata.name}')

echo $POD

Check the token file path exists:

kubectl -n monitoring exec -it $POD -- ls -l /var/run/secrets/kubernetes.io/serviceaccount/token
Optional: confirm it's mounted read-only (usually shown by mount options):

kubectl -n monitoring exec -it $POD -- mount | grep /var/run/secrets/kubernetes.io/serviceaccount

Q What the examiner checks

SA stats-monitor-sa has:

automountServiceAccountToken: false

Deployment stats-monitor mounts a projected volume named token

Token file is at:
/var/run/secrets/kubernetes.io/serviceaccount/token

Mount is readOnly: true

If label selector doesn't match (-l app=stats-monitor)
Use:

kubectl -n monitoring get pods

Then set:

POD=<paste-pod-name>

Question: 59

SIMULATION

Documentation Upgrading kubeadm clusters

You must connect to the correct host . Failure to do so may result in a zero score.

[candidate@base] $ ssh cks000034

Context

The kubeadm provisioned cluster was recently upgraded, leaving one node on a slightly older version due to workload compatibility concerns.

Task

Upgrade the cluster node compute-0 to match the version of the control plane node.

Use a command like the following to connect to the compute node:

[candidate@cks000034] $ ssh compute-0

Do not modify any running workloads in the cluster.

Do not forget to exit from the compute node once you have completed your tasks:

[candidate@icompute-e] $ exit

Answer: See the Explanation below for complete solution.

Explanation:

Below is the CKS / CKA exam-style, exact step-by-step solution for Upgrading a kubeadm worker node.

Follow in order, type exact commands, no extra actions.

QUESTION — Upgrade node compute-0 (EXAM MODE)

1) Connect to the correct host (control plane)

ssh cks000034

sudo -i

export KUBECONFIG=/etc/kubernetes/admin.conf

2) Identify the control plane Kubernetes version

This is the target version for compute-0.

www.atmicnetworks.com

kubectl get nodes

Example output:

NAME STATUS ROLES VERSION

control-plane Ready control-plane v1.27.4

compute-0 Ready <none> v1.26.6

Note the control-plane version

Example: v1.27.4

3) Drain the compute node (do NOT modify workloads manually) kubectl drain compute-0 --

ignore-daemonsets --delete-emptydir-data Wait until drain completes successfully.

4) SSH into the compute node

ssh compute-0

sudo -i

5) Check current kubeadm version on compute node

kubeadm version

6) Upgrade kubeadm to match control plane version

Replace 1.27.4 with the exact control-plane version you observed.

apt-get update

apt-get install -y kubeadm=1.27.4-00

Verify:

kubeadm version

7) Run kubeadm upgrade for the node

kubeadm upgrade node

Q This updates node-specific configs (NO workloads touched).

8) Upgrade kubelet and kubectl to the same version

apt-get install -y kubelet=1.27.4-00 kubectl=1.27.4-00

9) Restart kubelet

systemctl daemon-reload

systemctl restart kubelet

systemctl status kubelet --no-pager

10) Exit the compute node (IMPORTANT)

exit

11) Uncordon the compute node (back on control plane)

kubectl uncordon compute-0

12) Final verification

kubectl get nodes

Expected:

NAME STATUS VERSION

compute-0 Ready v1.27.4

Question: 60

SIMULATION

Documentation Deployments, Pods, bom Command Help bom-help

You must connect to the correct host. Failure to do so may result in a zero score.

www.atmicnetworks.com

www.atmicnetworks.com

[candidate@base] $ ssh cks000035

Task

The alpine Deployment in the alpine namespace has three containers that run different versions of the alpine image.

First, find out which version of the alpine image contains the libcrypto3 package at version 3.1.4-r5.

Next, use the pre-installed bom tool to create an SPDX document for the identified image version at /home/candidate/alpine.spdx.

You can find the bom tool documentation at bom.

Finally, update the alpine Deployment and remove the container that uses the idenfied image version.

The Deployment's manifest file can be found at /home/candidate/alpine-deployment.yaml.

Do not modify any other containers of the Deployment.

Answer: See the Explanation below for complete solution.

Explanation:

1) Connect to the correct host

ssh cks000035

sudo -i

export KUBECONFIG=/etc/kubernetes/admin.conf

2) List the 3 container names + images in the Deployment

kubectl -n alpine get deploy alpine -o jsonpath='{range .spec.template.spec.containers[*]}{.name}{"\t"}{.image}{"\n"}{end}'

You'll get 3 lines like:

c1 alpine:3.xx

c2 alpine:3.yy

c3 alpine:3.zz

3) Identify which alpine image has libcrypto3 at 3.1.4-r5

Fastest reliable method (since it's Alpine, just query apk inside each image):

Run these one-by-one for each image you saw in step 2:

docker run --rm <ALPINE_IMAGE_1> sh -c 'apk info -v libcrypto3 2>/dev/null | head -n1' docker run --rm <ALPINE_IMAGE_2> sh -c 'apk info -v libcrypto3 2>/dev/null | head -n1' docker run --rm <ALPINE_IMAGE_3> sh -c 'apk info -v libcrypto3 2>/dev/null | head -n1' Q The correct image is the one that prints exactly:

libcrypto3-3.1.4-r5

Note that full image tag, e.g.:

IMG=alpine:3.xx

4) Create SPDX document with bom for that identified image

(Use the identified image from step 3.)

bom generate --image $IMG --format spdx --output /home/candidate/alpine.spdx

Verify file exists:

ls -l /home/candidate/alpine.spdx

5) Remove ONLY the container that uses that image version

The manifest to edit is:

vi /home/candidate/alpine-deployment.yaml

In the spec.template.spec.containers: list, find the container entry whose image: equals the identified $IMG, and delete that one container block only (name/image/ports/etc for that container).

Save:

:wq

6) Apply the updated Deployment (do not change other containers)

kubectl apply -f /home/candidate/alpine-deployment.yaml

Wait rollout:

kubectl -n alpine rollout status deployment/alpine

7) Verify only 2 containers remain

kubectl -n alpine get deploy alpine -o jsonpath='{range .spec.template.spec.containers[*]}{.name}{"\t"}{.image}{"\n"}{end}'

You should now see 2 lines, and the $IMG line should be gone.

If bom generate ... errors (quick fix)

Check exact syntax on that system:

bom --help

bom generate --help

Then rerun with the flags it expects, keeping:

image = $IMG

output = /home/candidate/alpine.spdx

format = spdx

www.atmicnetworks.com

Question: 61
SIMULATION

Documentation

Deployment, Pod Security Admission, Pod Security Standards

You must connect to the correct host . Failure to do so may result in a zero score.

[candidate@base] $ ssh cks000036

Context

For compliance, all user namespaces enforce the restricted Pod Security Standard .

Task

The confidential namespace contains a Deployment that is not compliant with the restricted Pod Security Standard . Thus, its Pods can not be scheduled.
Modify the Deployment to be compliant and verify that the Pods are running.

The Deployment's manifest file can be found at /home/candidate/nginx-unprivileged.yaml.

Answer: See the Explanation below for complete solution.

Explanation:

1) Connect to the correct host

www.atmicnetworks.com

ssh cks000036

sudo -i

export KUBECONFIG=/etc/kubernetes/admin.conf

2) Confirm the failing Pods + see the PSA error (fast)

kubectl -n confidential get deploy

kubectl -n confidential get pods

kubectl -n confidential describe deploy <deployment-name> | sed -n '/Events/,$p'

(You'll usually see "violates PodSecurity 'restricted' ..." with the exact missing fields.)

3) Edit the provided manifest

vi /home/candidate/nginx-unprivileged.yaml

You must ensure the Pod template becomes compliant. Add/ensure the following exact blocks:

4) Add Pod-level securityContext (under spec.template.spec)

Find:

spec:

template:

spec:

Add this block under it (or merge if securityContext: already exists):

securityContext:

runAsNonRoot: true

runAsUser: 65535

seccompProfile:

type: RuntimeDefault

5) Add Container-level securityContext (under the nginx container)

www.atmicnetworks.com

Find:

containers:

- name: ...

image: ...

Under that container, add (or adjust) this exact block:

securityContext:

allowPrivilegeEscalation: false

readOnlyRootFilesystem: true

capabilities:

drop:

- ALL

If there are multiple containers, apply the same container securityContext to each one.

Save and exit:

:wq

6) Apply the manifest to the confidential namespace

kubectl -n confidential apply -f /home/candidate/nginx-unprivileged.yaml

Wait rollout:

kubectl -n confidential rollout status deployment/<deployment-name>

If you don't know the deployment name from the file, list: kubectl -n confidential get deploy

7) Verify Pods are running

kubectl -n confidential get pods -o wide

If still failing, show the exact PSA violation (this tells you what else to fix): kubectl -n confidential

describe pod <pod-name> | sed -n '/Events/,$p'

www.atmicnetworks.com

Quick “if it still fails” fixes (common restricted blockers)

Open the manifest again and ensure these are NOT set (or are removed/false): hostNetwork: true

hostPID: true hostIPC: true

any hostPort:

privileged: true

capabilities.add:

seccompProfile: Unconfined

runAsUser: 0 or runAsNonRoot: false

Then re-apply.

Minimal compliant result (what the grader expects)

Your Pod template should include:

seccompProfile: RuntimeDefault

runAsNonRoot: true (and a non-root UID like 65535)

container: allowPrivilegeEscalation: false

container: capabilities.drop: [ALL]

container: readOnlyRootFilesystem: true

Question: 62

SIMULATION

Documentation dockerd

You must connect to the correct host . Failure to do so may result in a zero score.

[candidate@base] $ ssh cks000037

www.atmicnetworks.com

Task

Perform the following tasks to secure the cluster node cks000037 :

Remove user developer from the docker group.

Do not remove the user from any other group.

Reconfigure and restart the Docker daemon to ensure that the socket file located at /var/run/docker.sock is owned by the group root.

Re-configure and restart the Docker daemon to ensure it does not listen on any TCP port.

After completing your work, ensure the Kubernetes cluster is healthy.

Answer: See the Explanation below for complete solution.

Explanation:

1) Connect to the correct host

ssh cks000037

sudo -i

2) Remove user developer from the docker group ONLY

2.1 Verify current groups (optional but fast)

id developer

2.2 Remove ONLY from docker group

gpasswd -d developer docker

2.3 Verify removal

id developer

Q docker should not appear; other groups must remain.

3) Reconfigure Docker to secure the socket and disable TCP

Docker config file:

vi /etc/docker/daemon.json

3.1 Set socket group to root and disable TCP listeners

Ensure the file contains exactly these relevant settings (merge with existing JSON if present):

{

"group": "root",

"hosts": ["unix:///var/run/docker.sock"]

}

Important:

"group": "root" → docker.sock owned by group root

"hosts" includes ONLY the unix socket (no tcp://)

If the file already exists with other keys, add/adjust only these keys and keep valid JSON (commas!).

Save and exit:

:wq

4) Restart Docker daemon systemctl daemon-reload

systemctl restart docker systemctl status docker --no-pager

5) Verify Docker socket ownership and permissions

ls -l /var/run/docker.sock

www.atmicnetworks.com

Expected:

srw-rw---- 1 root root ...

Q Owner: root
Q Group: root

6) Verify Docker is NOT listening on TCP

ss -lntp | grep docker

Expected:

No output (or nothing bound to TCP by dockerd)

Optional double-check:

ps aux | grep dockerd | grep -v grep

Ensure no -H tcp://... flags.

7) Ensure Kubernetes cluster is healthy

7.1 Check node and pods

export KUBECONFIG=/etc/kubernetes/admin.conf

kubectl get nodes

kubectl get pods -A

All nodes should be Ready, core pods Running.

Question: 63

SIMULATION

Documentation

Installing the Sidecar, PeerAuthentication, Deployments

You must connect to the correct host . Failure to do so may result in a zero score.

www.atmicnetworks.com

[candidate@base] $ ssh cks000041

Context

A microservices-based application using unencrypted Layer 4 (L4) transport must be secured with Istio.

Task

Perform the following tasks to secure an existing application's Layer 4 (L4) transport communication using Istio.

Istio is installed to secure Layer 4 (L4) communications.

You may use your browser to access Istio's documentation.

First, ensure that all Pods in the mtls namespace have the istio-proxy sidecar injected.

Next, configure mutual authentication in strict mode for all workloads in the mtls namespace.

Answer: See the Explanation below for complete solution.

Explanation:

Below is the CKS exam-ready, step-by-step solution for QUESTION 15. Follow exactly in this order. No extra changes.

QUESTION 15 — Istio mTLS (EXAM MODE)

1) Connect to the correct host

ssh cks000041

sudo -i

export KUBECONFIG=/etc/kubernetes/admin.conf

www.atmicnetworks.com

2) Ensure sidecar injection is enabled for the mtls namespace

2.1 Check current namespace labels

kubectl get ns mtls --show-labels

2.2 Enable automatic Istio sidecar injection

kubectl label namespace mtls istio-injection=enabled --overwrite

Verify:

kubectl get ns mtls --show-labels | grep istio-injection

Expected:

istio-injection=enabled

3) Ensure ALL Pods get the istio-proxy sidecar

Existing Pods will not get sidecars automatically.

You must restart workloads in the namespace.

3.1 Restart all Deployments in mtls

kubectl -n mtls rollout restart deployment

3.2 Verify Pods now have 2 containers (app + istio-proxy)

kubectl -n mtls get pods

Then check one Pod:

kubectl -n mtls get pod <pod-name> -o jsonpath='{.spec.containers[*].name}{"\n"}'

Expected output includes:

istio-proxy

4) Configure mutual TLS (mTLS) in STRICT mode

4.1 Create a PeerAuthentication for the mtls namespace

cat <<EOF | kubectl apply -f -

apiVersion: security.istio.io/v1beta1

kind: PeerAuthentication

www.atmicnetworks.com

metadata:

name: mtls-strict

namespace: mtls

spec:

mtls:

mode: STRICT

EOF

5) Verify mTLS policy is applied

kubectl -n mtls get peerauthentication

kubectl -n mtls describe peerauthentication mtls-strict

Expected:

Mode: STRICT

6) Final verification (exam confidence check)

6.1 Confirm all Pods are Running

kubectl -n mtls get pods

6.2 Confirm sidecar injection everywhere

kubectl -n mtls get pods -o jsonpath='{range .items[*]}{.metadata.name}{" ->
"}{.spec.containers[*].name}{"\n"}{end}'

Each line must include istio-proxy.

Question: 64

SIMULATION

Documentation Secrets, TLS Secrets, Volumes

You must connect to the correct host . Failure to do so may result in a zero score.

www.atmicnetworks.com

[candidate@base] $ ssh cks000m40

Path

Key

Context

You must complete securing access to a web server using SSL files stored in a TLS Secret .

Task

Create a TLS Secret named clever-cactus in the clever-cactus namespace for an existing Deployment named clever-cactus.

Use the following SSL files:

File

Certificate /home/candidate/clever-cactus/web.k8s.local.crt

/home/candidate/clever-cactus/web.k8s.local.key

The Deployment is already configured to use the TLS Secret.

Do not modify the existing Deployment.

Failure to do so may result in a reduced score.

Answer: See the Explanation below

www.atmicnetworks.com

for complete solution.

Explanation:

1) Connect to the correct host

ssh cks000m40

sudo -i

export KUBECONFIG=/etc/kubernetes/admin.conf

2) Verify namespace exists (quick check) kubectl get ns clever-cactus

3) Verify certificate and key files exist

ls -l /home/candidate/clever-cactus/web.k8s.local.crt

ls -l /home/candidate/clever-cactus/web.k8s.local.key Both files must exist.

4) Create the TLS Secret (THIS IS THE MAIN TASK)

Create a TLS Secret named clever-cactus in namespace clever-cactus:

kubectl -n clever-cactus create secret tls clever-cactus \

--cert=/home/candidate/clever-cactus/web.k8s.local.crt \

--key=/home/candidate/clever-cactus/web.k8s.local.key

! Do NOT use apply

! Do NOT edit the Deployment

5) Verify the Secret

kubectl -n clever-cactus get secret clever-cactus

Expected type:

kubernetes.io/tls

Optional detail check:

kubectl -n clever-cactus describe secret clever-cactus

www.atmicnetworks.com

You should see:

tls.crt

tls.key

6) (Optional) Confirm Pods are running

Since the Deployment is already configured to use the Secret, Pods should now work. kubectl -n clever-cactus get pods