Free Questions for CKS by actualtestdumps

Shared by Weiss on 18-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

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:

1.2.7authorization-modeargument is not set toAlwaysAllow FAIL

1.2.8authorization-modeargument includesNode FAIL

1.2.7authorization-modeargument includesRBAC FAIL

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

4.2.1 Ensure that theanonymous-auth argumentis set to false FAIL

4.2.2authorization-modeargument is not set to AlwaysAllow FAIL (UseWebhookautumn/authz where possible)

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

2.2 Ensure that theclient-cert-authargument is set to true

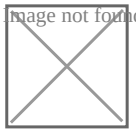
Options:

A) 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
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

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



Answer:

A

Question 2

Question Type: **MultipleChoice**

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 namedweb-podrunning in the namespacedatabase.

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

2. Create a new Role namedtest-role-2in the namespacedatabase, which only allows performingupdateoperations, only on resources of typestatuefulsets.

3. Create a new RoleBinding namedtest-role-2-bindbinding the newly created Role to the Pod's ServiceAccount.

Note: Don't delete the existing RoleBinding.

Options:

A) Explanation:

$k edit role test-role -n database

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

creationTimestamp: '2021-06-04T11:12:23Z'

name: test-role

namespace: database

resourceVersion: '1139'

selfLink: /apis/rbac.authorization.k8s.io/v1/namespaces/database/roles/test-role

uid: 49949265-6e01-499c-94ac-5011d6f6a353

rules:

- apiGroups:

- ''

resources:

- pods

verbs:

- * # Delete

- get # Fixed

$k create role test-role-2 -n database --resource statefulset --verb update

$k create rolebinding test-role-2-bind -n database --role test-role-2 --serviceaccount=database:test-sa

Explanation

[desk@cli]$k get pods -n database

NAME READY STATUS RESTARTS AGE LABELS

web-pod 1/1 Running 0 34s run=web-pod

[desk@cli]$k get roles -n database

test-role

[desk@cli]$k edit role test-role -n database

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

creationTimestamp: '2021-06-13T11:12:23Z'

name: test-role

namespace: database

resourceVersion: '1139'

selfLink: /apis/rbac.authorization.k8s.io/v1/namespaces/database/roles/test-role

uid: 49949265-6e01-499c-94ac-5011d6f6a353

rules:

- apiGroups:

- ''

resources:

- pods

verbs:

- '*' # Delete this

- get # Replace by this

[desk@cli]$k create role test-role-2 -n database --resource statefulset --verb update

role.rbac.authorization.k8s.io/test-role-2 created

[desk@cli]$k create rolebinding test-role-2-bind -n database --role test-role-2 --serviceaccount=database:test-sa
rolebinding.rbac.authorization.k8s.io/test-role-2-bind created

Reference:https://kubernetes.io/docs/reference/access-authn-authz/rbac/
role.rbac.authorization.k8s.io/test-role-2 created
[desk@cli]$k create rolebinding test-role-2-bind -n database --role test-role-2 --serviceaccount=database:test-sa
rolebinding.rbac.authorization.k8s.io/test-role-2-bind created
[desk@cli]$k create role test-role-2 -n database --resource statefulset --verb update
role.rbac.authorization.k8s.io/test-role-2 created

[desk@cli]$k create rolebinding test-role-2-bind -n database --role test-role-2 --serviceaccount=database:test-sa
rolebinding.rbac.authorization.k8s.io/test-role-2-bind created

Reference:https://kubernetes.io/docs/reference/access-authn-authz/rbac/

Answer:

A

Question 3

Question Type: MultipleChoice

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

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

Task:

Create a RuntimeClass namednot-trustedusing the prepared runtime handler namesrunsc.

Update all Pods in the namespace server to run onnewruntime.

Options:

A) Explanation:
Find all the pods/deployment and edit runtimeClassName parameter to not-trusted under spec
[desk@cli] $k edit deploy nginx
spec:
runtimeClassName: 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
nginx 3/3 11 3 5m
[desk@cli] $k edit deploy nginx
Image not found or type unknown

Answer:

A

Question 4

Question Type: MultipleChoice

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 nameddeny-networkin the namespacetestfor all traffic of type Ingress + Egress

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

Apply the newly createddefault-denyNetworkPolicy to all Pods running in namespacetest.

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

Options:

A) 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: {}
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/

Answer:

A

Explanation:

master1 $ k apply -f netpol1.yaml Reference: https://kubernetes.io/docs/concepts/services-networking/network-policies/

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/

master1 $ k apply -f netpol1.yaml Reference: https://kubernetes.io/docs/concepts/services-networking/network-policies/

Question 5

Question Type: MultipleChoice

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 namespacenato.

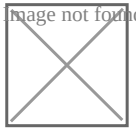
Look for images withHighorCriticalseverity vulnerabilities and delete the Pods that use those images.

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

Options:

A) Explanation:
[controlplane@cli] $k get pods -n nato -o yaml | grep 'image: '

[controlplane@cli] $trivy image <image-name>
[controlplane@cli] $k delete pod <vulnerable-pod> -n nato
[desk@cli] $ssh controlnode
[controlplane@cli] $k get pods -n nato
NAME READY STATUS RESTARTS AGE
alohmora 1/1 Running 0 3m7s
c3d3 1/1 Running 0 2m54s
neon-pod 1/1 Running 0 2m11s
thor 1/1 Running 0 58s
[controlplane@cli] $k get pods -n nato -o yaml | grep 'image: '
Image not found or type unknown



[controlplane@cli] $k delete pod thor -n nato
[controlplane@cli] $k delete pod neon-pod -n nato

Reference:https://github.com/aquasecurity/trivy
[controlplane@cli] $k delete pod neon-pod -n nato

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

Answer:

A

Question 6

Question Type: MultipleChoice

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 namedadamin thesafenamespace.

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 namesnewsecretin thesafenamespace, with the following content:

Username:dbadmin

Password:moresecurepas

Finally, create a new Pod that has access to the secretnewsecretvia a volume:

Namespace: safe

Pod name: mysecret-pod

Container name: db-container

Image: redis

Volume name: secret-vol

Mount path: /etc/mysecret

Options:

A) Explanation:

1. Get the secret, decrypt it & save in files

k get secret adam -n safe -o yaml

2. Create new secret using --from-literal

[desk@cli] $k create secret generic newsecret -n safe --from-literal=username=dbadmin --from-literal=password=moresecurepass

3. Mount it as volume of db-container of mysecret-pod

Explanation

Image not found or type unknown

Image not found or type unknown

[desk@cli] $k create secret generic newsecret -n safe --from-literal=username=dbadmin --from-literal=password=moresecurepass
secret/newsecret created
[desk@cli] $vim /home/certs_masters/secret-pod.yaml
apiVersion: v1
kind: Pod
metadata:
name: mysecret-pod
namespace: safe
labels:
run: mysecret-pod
spec:
containers:
- name: db-container
image: redis
volumeMounts:
- name: secret-vol
mountPath: /etc/mysecret
readOnly: true
volumes:

- name: secret-vol
secret:
secretName: newsecret
[desk@cli] $k apply -f /home/certs_masters/secret-pod.yaml
pod/mysecret-pod created
[desk@cli] $k exec -it mysecret-pod -n safe -- cat /etc/mysecret/username
dbadmin
Image not found or type unknown

[desk@cli] $k exec -it mysecret-pod -n safe -- cat /etc/mysecret/password
moresecurepas
Image not found or type unknown

Answer:

A

Question 7

Question Type: MultipleChoice

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 Podtomcat.

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:

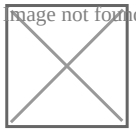
[timestamp],[uid],[processName]

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

Options:

A) 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
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
Image not found or type unknown



Answer:

A

Question 8

Question Type: MultipleChoice

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

Options:

A) 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
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/
pod/frontend created
[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/

Answer:

A