List of questions found in the CKS exam and in the exam dump (48 questions)

Summary

[Question N° 1: Creation SA, Role and Rolebinding 2](#_Toc158159688)

[Question N° 2: CIS-Benchmark 2](#_Toc158159689)

[Question N° 3: PodSecurityPolicy, SA, ClusterRole, clusterrolebinding 3](#_Toc158159690)

[Question N° 4: Creation SA, Role with less priv. and rolebind. 3](#_Toc158159691)

[Question N° 5: Audit log 3](#_Toc158159692)

[Question N° 6: Analyze Dockerfile 3](#_Toc158159693)

[Question N° 7: RuntimeClass 4](#_Toc158159694)

[Question N° 8: NetworkPolicy 4](#_Toc158159695)

[Question N° 9: Admission Controller Plugin, ImagePolicyWebhook 4](#_Toc158159696)

[Question N° 10: AppArmor Profile 4](#_Toc158159697)

[Question N° 11: Network Policy 4](#_Toc158159698)

[Question N° 12: Manage a secret 4](#_Toc158159699)

[Question N° 13: Trivy Image 5](#_Toc158159700)

[Question N° 14: Falco 5](#_Toc158159701)

[Question N° 15: CSR and CRT 5](#_Toc158159702)

[Question N° 16: CSI Benchmark violation 5](#_Toc158159703)

[Question No: 17 6](#_Toc158159704)

[Question No: 18 6](#_Toc158159705)

[Question No: 19 6](#_Toc158159706)

[Question No: 20 6](#_Toc158159707)

[Question No: 21 6](#_Toc158159708)

[Question No: 22 7](#_Toc158159709)

[Question No: 23 7](#_Toc158159710)

[Question No: 24 7](#_Toc158159711)

[Question No: 25 7](#_Toc158159712)

[Question No: 26 7](#_Toc158159713)

[Question No: 27 7](#_Toc158159714)

[Question No: 28 8](#_Toc158159715)

[Question No: 29 8](#_Toc158159716)

[Question No: 30 8](#_Toc158159717)

[Question No: 31 8](#_Toc158159718)

[Question No: 32 9](#_Toc158159719)

[Question No: 33 9](#_Toc158159720)

[Question No: 34 9](#_Toc158159721)

[Question No: 35 9](#_Toc158159722)

[Question No: 36 10](#_Toc158159723)

[Question No: 37 10](#_Toc158159724)

[Question No: 38 10](#_Toc158159725)

[Question No: 39 11](#_Toc158159726)

[Question No: 40 11](#_Toc158159727)

[Question No: 41 11](#_Toc158159728)

[Question No: 42 11](#_Toc158159729)

[Question No: 43 11](#_Toc158159730)

[Question No: 44 12](#_Toc158159731)

[Question No: 45 12](#_Toc158159732)

[Question No: 46 12](#_Toc158159733)

[Question No: 47 12](#_Toc158159734)

[Question No: 48 12](#_Toc158159735)

# Question N° 1: Creation SA, Role and Rolebinding

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.

# Question N° 2: CIS-Benchmark

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.

# Question N° 3: PodSecurityPolicy, SA, ClusterRole, clusterrolebinding

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.

# Question N° 4: Creation SA, Role with less priv. and rolebind.

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.

# Question N° 5: Audit log

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

# Question N° 6: Analyze Dockerfile

Analyze and edit the given Dockerfile

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

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.

# Question N° 7: RuntimeClass

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

# Question N° 8: NetworkPolicy

Create a NetworkPolicy named pod-access to restrict access to Pod users-service running in namespace dev-team. Only allow the following Pods to connect to Pod users-service: Pods in the namespace qa

Pods with label environment: testing, in any namespace.

# Question N° 9: Admission Controller Plugin, ImagePolicyWebhook

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.

# Question N° 10: AppArmor Profile

On the Cluster worker node, enforce the prepared AppArmor profile

#include <tunables/global>

profile nginx-deny flags=(attach\_disconnected) {

  #include <abstractions/base>

  file,

  # Deny all file writes.

  deny */\*\* w,*

*}*

*EOF*

Edit the prepared manifest file to include the AppArmor profile. Finally, apply the manifests files and create the Pod specified on it. Verify: Try to make a file inside the directory which is restricted.

# Question N° 11: Network Policy

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

# Question N° 12: Manage a secret

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

c. 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.

# Question N° 13: Trivy Image

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

# Question N° 14: Falco

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:

The following example shows a properly formatted incident file:

# Question N° 15: CSR and CRT

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.

# Question N° 16: CSI Benchmark violation

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.

# Question No: 17

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 psprole to the created SA psp-sa.

# Question No: 18

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).

# Question No: 19

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 services.

# Question No: 20

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.

# Question No: 21

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.

# Question No: 22

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.

# Question No: 23

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"

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.

# Question No: 24

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.

# Question No: 25

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.

# Question No: 26

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]

# Question No: 27

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.

# Question No: 28

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:

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

# Question No: 29

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.

# Question No: 30

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

# Question No: 31

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.

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

# Question No: 32

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.

# Question No: 33

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

# Question No: 34

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

# Question No: 35

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.

# Question No: 36

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

Pod name: mysecret-pod

Container name: db-container

Image: redis

Volume name: secret-vol

Mount path: /etc/mysecret

# Question No: 37

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.

# Question No: 38

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

# Question No: 39

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

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

# Question No: 40

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.

# Question No: 41

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

# Question No: 42

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).

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

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

# Question No: 43

Your organization’s security policy includes:

ServiceAccounts must not automount API credentials

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.

1. Create a new ServiceAccount named frontend-sa in the existing namespace q a. 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

# Question No: 44

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.

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:

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

# Question No: 45

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.

# Question No: 46

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.

# Question No: 47

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/bestpractice issues.

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.

# Question No: 48

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.