最新的CKS認證考試的題目與答案

P.S. NewDumps在Google Drive上分享了免費的、最新的CKS考試題庫：https://drive.google.com/open?id=11D7H6Gyvfl_QRSQX4GXoK0tE4PVFmC37

將NewDumps的產品加入購物車吧！你將以100%的信心去參加考試，一次性通過Linux Foundation CKS 認證考試，你將不會後悔你的選擇的。

擁有Linux Foundation CKS認證可以評估你在公司的價值和能力，但是通過這個考試是比較困難的。而CKS考題資料能幫考生掌握考試所需要的知識點，擁有良好的口碑，只要你選擇Linux Foundation CKS考古題作為你的考前復習資料，你就會相信自己的選擇不會錯。在您購買Linux Foundation CKS考古題之前，我們所有的題庫都有提供對應免費試用的demo，您覺得適合在購買，這樣您可以更好的了解我們產品的品質。

>> CKS套裝 <<

CKS熱門題庫 & CKS認證考試解析

NewDumps的CKS資料不僅能讓你通過考試，還可以讓你學到關於CKS考試的很多知識。NewDumps的考古題把你應該要掌握的技能全都包含在試題中，這樣你就可以很好地提高自己的能力，並且在工作中更好地應用它們。NewDumps的CKS考古題絕對是你準備考試並提高自己技能的最好的選擇。你要相信NewDumps可以給你一個美好的未來。

Linux Foundation CKS（Certified Kubernetes Security Specialist）認證考試是一個專業的認證計劃，旨在驗證個人在保障 Kubernetes 部署方面的專業知識。Kubernetes 是一個流行的開源平台，用於容器編排和管理，確保其安全性至關重要。CKS 認證考試是專業人士展示他們在保障 Kubernetes 環境方面的知識和經驗的一種方式。

CKS 考試旨在為有經驗的 Kubernetes 管理員和安全專家提供，該等人員負責保護 Kubernetes 環境的安全。考試涵蓋了廣泛的主題，包括 Kubernetes 集群設置、身份驗證和授權、網絡安全、存儲安全和容器安全等。候選人將在能否識別和減輕安全風險、實施安全策略、配置安全功能和審計 Kubernetes 環境方面進行考試。通過 CKS 考試需要對 Kubernetes 安全原則和實踐有深入的了解，以及在保護 Kubernetes 環境方面具有實踐經驗。

最新的 Kubernetes Security Specialist CKS 免費考試真題 (Q52-Q57):

問題 #52
SIMULATION

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 duster-admin 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.

答案：

解題說明：
See the Explanation below
Explanation:

問題 #53
You are managing a Kubernetes cluster running on AWS and need to assess the security configuration of the kubelet service against the CIS Kubernetes Benchmark v1 -7.1. You suspect that the '--cgroup-driver' flag is not properly configured, which could potentially expose the cluster to security vulnerabilities. Describe how you would use 'kubectl' to audit the current kubelet configuration and then determine the appropriate configuration tor the '-cgroup-driver' flag based on the CIS benchmark guidance. Assume that the kubelet service is running in a containerized environment.

答案：

解題說明：
Solution (Step by Step) :
1. Audit the kubelet configuration:
- Execute the following command to retrieve the kubelet configuration:
bash
kubectl get nodes -o jsonpath='{.items0.status.nodelnfo.kubeletVersion}'
- This command will output the kubelet version, which can be used to identifry the specific version of the CIS Kubernetes Benchmark that applies.
- Use 'kubectl describe node to retrieve the kubelet configuration for the specific node.
2. Review the CIS Benchmark guidance:
- Refer to the CIS Kubernetes Benchmark v1 -7.1 document for the specific guidance on the '--cgroup-driver' flag. The benchmark typically recommends using a specific 'cgroup-driver' value depending on the Kubernetes version and the underlying operating systenm
- For example, on a Kubernetes cluster running on AWS, the CIS bencnmark may recommend using the 'systemd' cgroup driver.
3. Determine the current kubelet configuration:
- Check the output of 'kubectl describe node for the value of the flag.
- This will show you the current configuration of the '-cgroup-driver' flag for the kubelet
5. Update the kubelet configuratiom
- Update the kubelet configuration for each node in your cluster to reflect the CIS benchmark recommendation. This may involve editing the kubelet configuration file or using a tool such as kubeadm or kubectl to modify the kubelet configuration.
6. Verify the changes:
- Run the audit commands again to verify that the kubelet configuration has been updated as expected.

問題 #54
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

答案：

解題說明：
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:
+ - 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
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'
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
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 --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: {}

問題 #55
Explain the role of security contexts in Kubernetes and how you would use them to mitigate potential security risks associated with container images.

答案：

解題說明：
Solution (Step by Step) :
1. understanding Security Contexts:
- Security Contexts in Kubernetes define the security attributes of a container, controlling its access to system resources and capabilities. They allow
you to enforce security policies and mitigate risks related to container images.
2. Key Security Context Settings:
- runASUser: Specifies the user ID under which the container will run. This can restrict access to files and resources that the container user might not need.
- runAsGroup: Similar to 'runAsUser, but for the group ID.
- fsGroup: Controls file system permissions. By setting this, you can grant specific access to certain files and directories.
- readOnlyRootFilesystem: Prevents the container from modifying the root file system
- privileged: Grants the container full root privileges. It should be avoided whenever possible.
- allowPfivilegeEscalatiom Controls whether the container can elevate its privileges.
- capabilities: Defines the Linux capabilities that the container is allowed to use. This can restrict access to specific system resources and operations.
- seLinuxOptions: Controls the benavior of the containers SELinux context. This can be used to enforce additional security policies based on SELinux.
3. Using Security Contexts for Image Security:
- Restricting Privileges: Set 'runAsUser', 'runAsGroup', 'privileged' , and 'allowPrivilegeEscalation' to limit the privileges of a container.
- Controlling File System Access: Utilize 'tsGroup' and readOnlyRootFilesystem' to restrict the containers ability to modify files and directories, minimizing the impact of potential vulnerabilities.
- Limiting Capabilities: Use the 'capabilities' field to selectively enable only the capabilities that the container needs to run. This can prevent malicious
code from accessing sensitive system resources.
- Enforcing SELinux Policies: Configure 'seLinuxOptionS to enforce stricter security policies that are aligned with your overall security requirements.
4. Example Security Context in Deployment YAML:
5. Best Practices: - Least Privilege Principle: Apply the least privilege principle to security contexts. Only grant containers the resources and capabilities they require. - Security Context Constraints: Define security context constraints (SCC) tor your cluster. SCCS entorce security policies across all pods. - Regular Auditing: Periodically review and adjust security context settings to ensure they align with your evolving security requirements. - Consider Security Tools: Use tools like Kubernetes Security Posture Management (KSPM) and security scanning solutions to help enforce and monitor security context configurations.

問題 #56
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.

答案：

解題說明：

問題 #57

.....