CKS최신업데이트시험덤프최신시험기출문제

참고: Pass4Test에서 Google Drive로 공유하는 무료, 최신 CKS 시험 문제집이 있습니다: https://drive.google.com/open?id=1GN3rO_QadoKf6W6285h0STrFz9pDCmp4

Pass4Test는 아주 믿을만하고 서비스 또한 만족스러운 사이트입니다. 만약 CKS시험실패 시 우리는 100% 덤프비용 전액환불 해드립니다.그리고 시험을 패스하여도 우리는 일 년 동안 무료업뎃을 제공합니다.

CKS 시험은 2 시간 이내에 완료 해야하는 15-20 개의 성능 기반 작업으로 구성된 엄격한 평가입니다. 시험은 온라인으로 수행되며 응시자는 명령 줄 도구 및 Kubernetes API 객체에 대한 지식뿐만 아니라 Kubernetes 클러스터에 액세스해야합니다. 인증은 2 년 동안 유효하며 재 인증 시험을 통과하거나 지속적인 교육 학점을 취득하여 갱신 할 수 있습니다.

Linux Foundation Certified Kubernetes Security Specialist (CKS) 시험은 Kubernetes 보안 전문가의 전문 지식을 검증하는 인증입니다. 이 인증 시험은 안전한 Kubernetes 클러스터를 설계, 배포 및 관리 할 수있는 전문가의 지식, 기술 및 능력을 테스트하도록 설계되었습니다. CKS 인증 시험은 응시자가 Kubernetes 보안 원칙 및 모범 사례에 대한 사전 지식과 경험을 갖도록 요구하는 고급 레벨 인증입니다.

>> CKS최신 업데이트 시험덤프 <<

퍼펙트한 CKS최신 업데이트 시험덤프 최신 덤프문제

우리는 여러분이 시험패스는 물론 또 일년무료 업데이트서비스를 제공합니다.만약 시험에서 실패했다면 우리는 덤프비용전액 환불을 약속 드립니다.하지만 이런 일은 없을 것입니다.우리는 우리덤프로 100%시험패스에 자신이 있습니다. 여러분은 먼저 우리 Pass4Test사이트에서 제공되는Linux Foundation인증CKS시험덤프의 일부분인 데모 즉 문제와 답을 다운받으셔서 체험해보실 수 잇습니다.

최신 Kubernetes Security Specialist CKS 무료샘플문제 (Q46-Q51):

질문 # 46
You are running a web application in a Kubernetes cluster using a Deployment named 'web-apps. The application is vulnerable to a known CVE that can be exploited through tne web server. You need to implement a security policy to prevent pods from accessing the vulnerable web server port.

정답：

설명：
Solution (Step by Step) :
1. Identity the vulnerable port:
- For this example, assume the vulnerable port is 8080.
2. Create a Securitycontext for the web server:
3. Apply the updated Deployment: bash kubectl apply -f web-app-deployment.yaml - The 'securityContext' is used to restrict the capabilities of the container. - 'drop: ["NET BIND SERVICET prevents the container from binding to ports below 1024 (including port 8080). - This policy will prevent pods from accessing the vulnerable web server port and mitigate the CVE. Important Notes: -

You can adjust the 'drop' list to restrict other capabilities as needed. - You might need to redeploy the web application with a different port that is not restricted-

질문 # 47
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
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

- A. Send us the Feedback on it.

정답：A

질문 # 48
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

정답：

설명：
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: {}

질문 # 49
SIMULATION
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.
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.

정답：

설명：

질문 # 50
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.

정답：

설명：
k get pods -n prod
k get pod <pod-name> -n prod -o yaml | grep -E 'privileged|ReadOnlyRootFileSystem' Delete the pods which do have any of these 2 properties privileged:true or ReadOnlyRootFileSystem: false
[desk@cli]$ k get pods -n prod
NAME READY STATUS RESTARTS AGE
cms 1/1 Running 0 68m
db 1/1 Running 0 4m
nginx 1/1 Running 0 23m
[desk@cli]$ k get pod nginx -n prod -o yaml | grep -E 'privileged|RootFileSystem'
{"apiVersion":"v1","kind":"Pod","metadata":{"annotations":{},"creationTimestamp":null,"labels":
{"run":"nginx"},"name":"nginx","namespace":"prod"},"spec":{"containers":[{"image":"nginx","name":"nginx","resources":
{},"securityContext":{"privileged":true}}],"dnsPolicy":"ClusterFirst","restartPolicy":"Always"},"status":{}} f:privileged: {} privileged: true
[desk@cli]$ k delete pod nginx -n prod

[desk@cli]$ k get pod db -n prod -o yaml | grep -E 'privileged|RootFilesystem'
[desk@cli]$ k delete pod cms -n prod Reference: https://kubernetes.io/docs/concepts/policy/pod-security-policy/ https://cloud.google.com/architecture/best-practices-for-operating-containers Reference:
[desk@cli]$ k delete pod cms -n prod Reference: https://kubernetes.io/docs/concepts/policy/pod-security-policy/ https://cloud.google.com/architecture/best-practices-for-operating-containers

질문 # 51
.....