Exam Questions KCSA

Kubernetes and Cloud Native Security Associate (KCSA)

https://www.2passeasy.com/dumps/KCSA/

NEW QUESTION 1
Which standard approach to security is augmented by the 4C??s of Cloud Native security?

A. Zero Trust
B. Least Privilege
C. Defense-in-Depth
D. Secure-by-Design

Answer: C

Explanation:

The 4C??s model (Cloud, Cluster, Container, Code) is presented in the official Kubernetes documentation as alayeredmodel that explicitly maps todefense-in-depth.

Exact extracts from Kubernetes docs(security overview):

??The 4C??s of Cloud Native Security are Cloud, Clusters, Containers, and Code.??

??You can think of the 4C??s asa layered approach to security; applying security measures at each layer reduces risk.??

??This layered approach is commonly known asdefense in depth.??

References:
Kubernetes Docs — Security overview #The 4C??s of Cloud Native Security: https://kubernetes.io/docs /concepts/security/overview/#the-4cs-of-cloud-native-security

NEW QUESTION 2
Given a standard Kubernetes cluster architecture comprising a single control plane node (hosting bothetcdand the control plane as Pods) and three worker nodes, which of the following data flows crosses atrust boundary?

A. From kubelet to Container Runtime
B. From kubelet to API Server
C. From kubelet to Controller Manager
D. From API Server to Container Runtime

Answer: B

Explanation:

Trust boundariesexist where data flows between different security domains.

In Kubernetes:

Communication between thekubelet (node agent)and theAPI Server (control plane)crosses the node-to-control-plane trust boundary.

(A) Kubelet to container runtime is local, no boundary crossing.

(C) Kubelet does not communicate directly with the controller manager.

(D) API server does not talk directly to the container runtime; it delegates to kubelet.

Therefore, (B) is the correct trust boundary crossing flow.

References:
CNCF Security Whitepaper – Kubernetes Threat Model: identifies node-to-control-plane communications (kubelet # API Server) as crossing trust boundaries.
Kubernetes Documentation – Cluster Architecture

NEW QUESTION 3
On a client machine, what directory (by default) contains sensitive credential information?

A. /etc/kubernetes/
B. $HOME/.kube
C. /opt/kubernetes/secrets/
D. $HOME/.config/kubernetes/

Answer: B

Explanation:

Thekubectlclient uses configuration from$HOME/.kube/configby default.

This file contains: cluster API server endpoint, user certificates, tokens, or kubeconfigs #sensitive credentials.

Exact extract (Kubernetes Docs – Configure Access to Clusters):

??By default, kubectl looks for a file named config in the $HOME/.kube directory. This file contains configuration information including user credentials.??

Other options clarified:

- A: /etc/kubernetes/ exists on nodes (control plane) not client machines.
- C: /opt/kubernetes/secrets/ is not a standard path.
- D: $HOME/.config/kubernetes/ is not where kubeconfig is stored by default.

References:
Kubernetes Docs — Configure Access to Clusters: https://kubernetes.io/docs/concepts/configuration/organize- cluster-access-kubeconfig/

NEW QUESTION 4
In a Kubernetes environment, what kind of Admission Controller can modify resource manifests when applied to the Kubernetes API to fix misconfigurations automatically?

A. ValidatingAdmissionController
B. PodSecurityPolicy
C. MutatingAdmissionController
D. ResourceQuota

Answer: C

Explanation:

- Kubernetes Admission Controllers can eithervalidateormutateincoming requests.
- MutatingAdmissionWebhook (Mutating Admission Controller):
- Canmodify or mutate resource manifestsbefore they are persisted in etcd.
- Used for automatic injection of sidecars (e.g., Istio Envoy proxy), setting default values, or fixing misconfigurations.
- ValidatingAdmissionWebhook (Validating Admission Controller):only allows/denies but doesnot change requests.
- PodSecurityPolicy:deprecated; cannot mutate requests.
- ResourceQuota:enforces resource usage, but does not mutate manifests.

Exact Extract:

- ??Mutating admission webhooks are invoked first, and can modify objects to enforce defaults.

Validating admission webhooks are invoked second, and can reject requests to enforce invariants.??
References:
Kubernetes Docs — Admission Controllers: https://kubernetes.io/docs/reference/access-authn-authz
/admission-controllers/
Kubernetes Docs — Admission Webhooks: https://kubernetes.io/docs/reference/access-authn-authz
/extensible-admission-controllers/

NEW QUESTION 5
Why does the defaultbase64 encodingthat Kubernetes applies to the contents of Secret resources provide inadequate protection?

A. Base64 encoding is vulnerable to brute-force attacks.
B. Base64 encoding relies on a shared key which can be easily compromised.
C. Base64 encoding does not encrypt the contents of the Secret, only obfuscates it.
D. Base64 encoding is not supported by all Secret Stores.

Answer: C

Explanation:

- Kubernetes stores Secret data asbase64-encoded stringsin etcd by default.
- Base64 is not encryption— it is a simple encoding scheme that merelyobfuscatesdata for transport and storage. Anyone with read access to etcd or the Secret manifest can easily decode the value back to plaintext.
- For actual protection, Kubernetes supportsencryption at rest(via encryption providers) and external Secret management (Vault, KMS, etc.).

[References:, Kubernetes Documentation – Secrets, CNCF Security Whitepaper – Data protection section: highlights that base64 encoding does not protect data and encryption at rest is recommended.,]

NEW QUESTION 6
In a Kubernetes cluster, what are the security risks associated with using ConfigMaps for storing secrets?

A. Storing secrets in ConfigMaps does not allow for fine-grained access control via RBAC.
B. Storing secrets in ConfigMaps can expose sensitive information as they are stored in plaintext and can be accessed by unauthorized users.
C. Using ConfigMaps for storing secrets might make applications incompatible with the Kubernetes cluster.
D. ConfigMaps store sensitive information in etcd encoded in base64 format automatically, which does not ensure confidentiality of data.

Answer: B

Explanation:

- ConfigMaps are explicitly not for confidential data.

Exact extract (ConfigMap concept):"A ConfigMap is an API object used to store non-confidential data in key-value pairs."

Exact extract (ConfigMap concept):"ConfigMaps are not intended to hold confidential data. Use a Secret for confidential data."

Why this is risky:data placed into a ConfigMap is stored as regular (plaintext) string values in the API and etcd (unless you deliberately use binaryData for base64 content you supply). That means if someone has read access to the namespace or to etcd/APIServer storage, they can view the values.

Secrets vs ConfigMaps (to clarify distractor D):

Exact extract (Secret concept):"By default, secret data is stored as unencrypted base64-encoded strings.You canenable encryption at restto protect Secrets stored in etcd."

This base64 behavior applies toSecrets, not to ConfigMap data. Thus optionDis incorrect for ConfigMaps.

About RBAC (to clarify distractor A):Kubernetesdoessupport fine-grained RBAC forbothConfigMaps and Secrets; the issue isn't lack of RBAC but that ConfigMaps arenotdesigned for confidential material.

About compatibility (to clarify distractor C):Using ConfigMaps for secrets doesn't make apps "incompatible"; it's simplyinsecureand against guidance.
[References:, Kubernetes Docs —ConfigMaps: https://kubernetes.io/docs/concepts/configuration/configmap/, Kubernetes Docs —Secrets: https://kubernetes.io/docs/concepts/configuration/secret/, Kubernetes Docs —Encrypting Secret Data at Rest: https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/, Note: The citations above are from the official Kubernetes documentation and reflect the stated guidance that ConfigMaps are fornon-confidentialdata, while Secrets (with encryption at rest enabled) are forconfidentialdata, and that the 4C's map todefense in depth.,]

NEW QUESTION 7
You want to minimize security issues in running Kubernetes Pods. Which of the following actions can help achieve this goal?

A. Sharing sensitive data among Pods in the same cluster to improve collaboration.
B. Running Pods with elevated privileges to maximize their capabilities.
C. Implement Pod Security standards in the Pod's YAML configuration.
D. Deploying Pods with randomly generated names to obfuscate their identities.

Answer: C

Explanation:
Pod Security Standards (PSS):
Kubernetes providesPod Security Admission (PSA)to enforce security controls based on policies.
Official extract: ??Pod Security Standards define different isolation levels for Pods. The standards focus on restricting what Pods can do and what they can access.??
The three standard profiles are:
Privileged: unrestricted (not recommended).
Baseline: minimal restrictions.
Restricted: highly restricted, enforcing least privilege.
Why option C is correct:
Applying Pod Security Standards in YAML ensures Pods adhere tobest practiceslike:
No root user.
Restricted host access.
No privilege escalation.
Seccomp/AppArmor profiles.
This directly minimizes security risks.
Why others are wrong:
A:Sharing sensitive data increases risk of exposure.
B:Running with elevated privileges contradicts least privilege principle.
D:Random Pod names donotcontribute to security.
[References:, Kubernetes Docs — Pod Security Standards: https://kubernetes.io/docs/concepts/security/pod-security-standards/, Kubernetes Docs — Pod Security Admission: https://kubernetes.io/docs/concepts/security/pod-security-admission/,]

NEW QUESTION 8
A container running in a Kubernetes cluster has permission to modify host processes on the underlying node.
What combination of privileges and capabilities is most likely to have led to this privilege escalation?

A. There is no combination of privileges and capabilities that permits this.
B. hostPID and SYS_PTRACE
C. hostPath and AUDIT_WRITE
D. hostNetwork and NET_RAW

Answer: B

Explanation:
hostPID:When enabled, the container shares the host??s process namespace ?? container can see and potentially interact with host processes.
SYS_PTRACE capability:Grants the container the ability to trace, inspect, and modify other processes (e.g., via ptrace).
Combination of hostPID + SYS_PTRACE allows a container toattach to and modify host processes, which is a direct privilege escalation.
Other options explained:
hostPath + AUDIT_WRITE:hostPath exposes filesystem paths but does not inherently allow process modification.
hostNetwork + NET_RAW:grants raw socket access but only for networking, not host process modification.

A:Incorrect — such combinationsdo exist(like B).
[References:, Kubernetes Docs — Configure a Pod to use hostPID: https://kubernetes.io/docs/tasks/configure-pod-container/share-process-namespace/, Linux Capabilities man page: https://man7.org/linux/man-pages/man7/capabilities.7.html,]

NEW QUESTION 9
Which of the following statements on static Pods is true?

A. The kubelet can run static Pods that span multiple nodes, provided that it has the necessary privileges from the API server.
B. The kubelet can run a maximum of 5 static Pods on each node.
C. The kubelet schedules static Pods local to its node without going through the kube-scheduler, making tracking and managing them difficult.
D. The kubelet only deploys static Pods when the kube-scheduler is unresponsive.

Answer: C

Explanation:
Static Podsare managed directly by thekubeleton each node.
They arenot scheduled by the kube-schedulerand always remain bound to the node where they are defined.
Exact extract (Kubernetes Docs – Static Pods):
??Static Pods are managed directly by the kubelet daemon on a specific node, without the API server. They do not go through the Kubernetes scheduler.??
Clarifications:
A: Static Pods do not span multiple nodes.
B: No hard limit of 5 Pods per node.
D: They are not a fallback mechanism; kubelet always manages them regardless of scheduler state.
References:
Kubernetes Docs — Static Pods: https://kubernetes.io/docs/tasks/configure-pod-container/static-pod/

NEW QUESTION 10
In a cluster that contains Nodes withmultiple container runtimesinstalled, how can a Pod be configured to be created on a specific runtime?

A. By using a command-line flag when creating the Pod.
B. By modifying the Docker daemon configuration.
C. By setting the container runtime as an environment variable in the Pod.
D. By specifying the container runtime in the Pod's YAML file.

Answer: D

Explanation:
Kubernetes supportsmultiple container runtimeson a node via theRuntimeClassresource.
To select a runtime, you specify the runtimeClassName field in thePod??s YAML manifest. Example:
apiVersion: v1
kind: Pod
metadata:
name: example
spec:
runtimeClassName: gvisor
containers:
- name: app
image: nginx
Incorrect options:
(A) You cannot specify container runtime through a kubectl command-line flag.
(B) Modifying the Docker daemon config does not direct Kubernetes Pods to a runtime.
(C) Environment variables inside a Pod spec do not control container runtimes.
References:
Kubernetes Documentation – RuntimeClass
CNCF Security Whitepaper – Workload isolation via different runtimes (e.g., gVisor, Kata) for enhanced security.

NEW QUESTION 10
What kind of organization would need to be compliant with PCI DSS?

A. Retail stores that only accept cash payments.
B. Government agencies that collect personally identifiable information.
C. Non-profit organizations that handle sensitive customer data.
D. Merchants that process credit card payments.

Answer: D

Explanation:
PCI DSS (Payment Card Industry Data Security Standard):applies to any entity thatstores, processes, or transmits cardholder data.
Exact extract (PCI DSS official summary):
"PCI DSS applies to all entities that store, process or transmit cardholder data (CHD) and /or sensitive authentication data (SAD)."
Therefore,merchants who process credit card paymentsmust comply.
Why others are wrong:
A: No card payments, so no PCI scope.
B: This falls underFISMA / NIST 800-53, not PCI DSS.
C: Non-profits may handle sensitive data, but PCI only applies if they processcredit cards.
References:
PCI Security Standards Council — PCI DSS Summary: https://www.pcisecuritystandards.org/pci_security/

NEW QUESTION 15
Which of the following is a valid security risk caused by having no egress controls in a Kubernetes cluster?

A. Denial of Service
B. Data exfiltration
C. Increased attack surface
D. Unauthorized access to external resources

Answer: B

Explanation:
Egress NetworkPoliciesrestrict outbound traffic from Pods.
Without egress restrictions, a compromised Pod could exfiltrate sensitive data (secrets, logs, customer data) to an attacker-controlled server.
Exact extract (Kubernetes Docs – Network Policies):
"Egress rules control outbound connections from Pods. Without such restrictions, compromised workloads can connect freely to external endpoints."
Other options clarified:
A: DoS is more about flooding, not egress absence.
C: ??Increased attack surface?? is vague but not the main risk.
D: True in a sense, but the precise and most common risk isdata exfiltration.
[References:, Kubernetes Docs — Network Policies: https://kubernetes.io/docs/concepts/services-networking/network-policies/,]

NEW QUESTION 19
Which of the following statements is true concerning the use ofmicroVMsover user-space kernel implementations for advanced container sandboxing?

A. MicroVMs allow for easier container management and orchestration than user-space kernel implementation.
B. MicroVMs offer higher isolation than user-space kernel implementations at the cost of a higher per-instance memory footprint.
C. MicroVMs provide reduced application compatibility and higher per-system call overhead than user-space kernel implementations.
D. MicroVMs offer lower isolation and security compared to user-space kernel implementations.

Answer: B

Explanation:
MicroVM-based runtimes(e.g., Firecracker, Kata Containers) use lightweight VMs to provide strong isolation between workloads.
Compared touser-space kernel implementations(e.g., gVisor), microVMs generally:
Offerhigher isolation and security(due to VM-level separation).
Come with ahigher memory and resource overhead per instancethan user-space approaches.
Incorrect options:
(A) Orchestration is handled by Kubernetes, not inherently easier with microVMs.
(C) Compatibility is typically better with microVMs, not worse.
(D) Isolation is stronger, not weaker.
[References:, CNCF Security Whitepaper – Workload isolation: microVMs vs. user-space kernel sandboxes., Kata Containers Project – isolation trade-offs.,]

NEW QUESTION 22
Which of the following statements regarding a container run with privileged: true is correct?

A. A container run with privileged: true within a cluster can access all Secrets used within that cluster.
B. A container run with privileged: true within a Namespace can access all Secrets used within that Namespace.
C. A container run with privileged: true on a node can access all Secrets used on that node.
D. A container run with privileged: true has no additional access to Secrets than if it were run with privileged: false.

Answer: D

Explanation:
Setting privileged: true grants a containerelevated access to the host node, including access to host devices, kernel capabilities, and the ability to modify the host.
However, Secrets in Kubernetes are not automatically exposedto privileged containers. Secrets are mounted into Pods only if explicitly referenced.
Thus, being privilegeddoes not grant additional access to Kubernetes Secretscompared to a non-privileged Pod.
The risk lies in node compromise: if a privileged container can take over the node, it could then indirectly gain access to Secrets (e.g., by reading kubelet credentials).
[References:, Kubernetes Documentation – Security Context, CNCF Security Whitepaper – Pod security context and privileged container risks.,]

NEW QUESTION 26
Which of the following snippets from a RoleBinding correctly associates user bob with Role pod-reader ?

A. subjects:- kind: Username: pod-readerapiGroup: rbac.authorization.k8s.ioroleRef:kind: Rolename: bobapiGroup: rbac.authorization.k8s.io
B. subjects:- kind: Username: bobapiGroup: rbac.authorization.k8s.ioroleRef:kind: Rolename: pod-readerapiGroup: rbac.authorization.k8s.io
C. subjects:- kind: Username: bobapiGroup: rbac.authorization.k8s.ioroleRef:kind: ClusterRolename: pod-readerapiGroup: rbac.authorization.k8s.io
D. subjects:- kind: Groupname: bobapiGroup: rbac.authorization.k8s.ioroleRef:kind: Rolename: pod-readerapiGroup: rbac.authorization.k8s.io

Answer: B

Explanation:
Kubernetes RBAC usesRoleBindingto grant permissions defined in aRoleto asubject(user, group, or service account) within a namespace. The official example shows binding user jane to Role pod-reader:
"A RoleBinding grants the permissions defined in a Role to a user or set of users??."
Example:
subjects:
- kind: User
name: jane
apiGroup: rbac.authorization.k8s.io
roleRef:

kind: Role
name: pod-reader
apiGroup: rbac.authorization.k8s.io
— Kubernetes docs, RBAC: RoleBinding and ClusterRoleBinding
OptionBmatches this pattern exactly, with name: bob as theUsersubject and roleRef pointing to theRolenamed pod-reader.
Aswaps the names (subject is pod-reader, role is bob) ?? incorrect.
Creferences aClusterRole, not aRole(the question asks for Role).
Duses kind: Group even though we need theUserbob.
[References:, Kubernetes Docs — Using RBAC Authorization ??RoleBinding and ClusterRoleBinding: https://kubernetes.io/docs/reference/access-authn-authz/rbac/#rolebinding-and-clusterrolebinding,]

NEW QUESTION 27
How can a user enforce thePod Security Standardwithout third-party tools?

A. Through implementing Kyverno or OPA Policies.
B. Use the PodSecurity admission controller.
C. It is only possible to enforce the Pod Security Standard with additional tools within the cloud native ecosystem.
D. No additional measures have to be taken to enforce the Pod Security Standard.

Answer: B

Explanation:
ThePodSecurity admission controller(built-in as of Kubernetes v1.23+) enforces the Pod Security Standards (Privileged, Baseline, Restricted).
Enforcement is namespace-scoped and configured throughnamespace labels.
Incorrect options:
(A) Kyverno/OPA are external policy tools (useful but not required).
(C) Not true, PodSecurity admission provides native enforcement.
(D) Enforcement requires explicit configuration, not automatic.
[References:, Kubernetes Documentation – Pod Security Admission, CNCF Security Whitepaper – Policy enforcement and admission control.,]

NEW QUESTION 31
What is the main reason an organization would use a Cloud Workload Protection Platform (CWPP) solution?

A. To protect containerized workloads from known vulnerabilities and malware threats.
B. To automate the deployment and management of containerized workloads.
C. To manage networking between containerized workloads in the Kubernetes cluster.
D. To optimize resource utilization and scalability of containerized workloads.

Answer: A

Explanation:
CWPP (Cloud Workload Protection Platform):As defined by Gartner and adopted across cloud
security practices, CWPPs are designed tosecure workloads(VMs, containers, serverless functions) in
hybrid and cloud environments.
They providevulnerability scanning, runtime protection, compliance checks, and malware detection.
Exact extract (Gartner CWPP definition):??Cloud workload protection platforms protect workloads regardless of location, including physical machines, VMs, containers, and serverless workloads. They provide vulnerability management, system integrity protection, intrusion detection and prevention, and malware protection.??
References:
Gartner: Cloud Workload Protection Platforms Market Guide (summary): https://www.gartner.com/reviews
/market/cloud-workload-protection-platforms
CNCF Security Whitepaper:https://github.com/cncf/tag-security

NEW QUESTION 36
By default, in a Kubeadm cluster, which authentication methods are enabled?

A. OIDC, Bootstrap tokens, and Service Account Tokens
B. X509 Client Certs, OIDC, and Service Account Tokens
C. X509 Client Certs, Bootstrap Tokens, and Service Account Tokens
D. X509 Client Certs, Webhook Authentication, and Service Account Tokens

Answer: C

Explanation:
In akubeadm cluster, by default the API server enables several authentication mechanisms:
X509 Client Certs: Used for authenticating kubelets, admins, and control-plane components.
Bootstrap Tokens: Temporary credentials used for node bootstrap/joining clusters.
Service Account Tokens: Used by workloads in pods to authenticate with the API server.
Exact extract (Kubernetes Docs – Authentication):
"Kubernetes uses client certificates, bearer tokens, an authenticating proxy, or HTTP basic auth to authenticate API requests."
"Bootstrap tokens are a simple bearer token that is meant to be used when creating new clusters or joining new nodes to an existing cluster."
"Service accounts are special accounts that provide an identity for processes that run in a Pod."
References:
Kubernetes Docs — Authentication: https://kubernetes.io/docs/reference/access-authn-authz/authentication/
Kubeadm — TLS Bootstrapping: https://kubernetes.io/docs/reference/access-authn-authz/bootstrap-tokens/

NEW QUESTION 41
Which of the following is a control for Supply Chain Risk Management according to NIST 800-53 Rev. 5?

A. Access Control
B. System and Communications Protection
C. Supply Chain Risk Management Plan
D. Incident Response

Answer: C

NEW QUESTION 44
.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual KCSA Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the KCSA Product From:

https://www.2passeasy.com/dumps/KCSA/

Money Back Guarantee

KCSA Practice Exam Features:

* KCSA Questions and Answers Updated Frequently

* KCSA Practice Questions Verified by Expert Senior Certified Staff

* KCSA Most Realistic Questions that Guarantee you a Pass on Your FirstTry

* KCSA Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year