Linux-Foundation

Exam Questions KCSA

Kubernetes and Cloud Native Security Associate (KCSA)

About Exambible

Your Partner of IT Exam

Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

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
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 4
What is the difference between gVisor and Firecracker?

A. gVisor is a user-space kernel that provides isolation and security for container
B. At the same time, Firecracker is a lightweight virtualization technology for creating and managing secure, multi-tenant container and function-as-a-service (FaaS) workloads.
C. gVisor is a lightweight virtualization technology for creating and managing secure, multi-tenant container and function-as-a-service (FaaS) workload
D. At the same time, Firecracker is a user-space kernel that provides isolation and security for containers.
E. gVisor and Firecracker are both container runtimes that can be used interchangeably.
F. gVisor and Firecracker are two names for the same technology, which provides isolation and security for containers.

Answer: A

Explanation:

gVisor:
Google-developed, implemented as auser-space kernelthat intercepts and emulates syscalls made by containers.
Providesstrong isolationwithout requiring a full VM.
Official docs: "gVisor is a user-space kernel, written in Go, that implements a substantial portion of the Linux system call interface."
Source: https://gvisor.dev/docs/

Firecracker:
AWS-developed,lightweight virtualization technologybuilt on KVM, used in AWS Lambda and Fargate.
Optimized for running secure, multi-tenant microVMs (MicroVMs) for containers and FaaS.
Official docs: "Firecracker is an open-source virtualization technology that is purpose-built for creating and managing secure, multi-tenant container and function-based services."
Source: https://firecracker-microvm.github.io/

Key difference:gVisor ?? syscall interception in userspace kernel (container isolation). Firecracker ?? lightweight virtualization with microVMs (multi-tenant security).
Therefore, optionAis correct.
[References:, gVisor Docs: https://gvisor.dev/docs/, Firecracker Docs: https://firecracker-microvm.github.io/,]

NEW QUESTION 5
Is it possible to restrict permissions so that a controller can only change the image of a deployment (without changing anything else about it, e.g., environment variables, commands, replicas, secrets)?

A. Yes, by granting permission to the /image subresource.
B. Not with RBAC, but it is possible with an admission webhook.
C. No, because granting access to the spec.containers.image field always grants access to the rest of the spec object.
D. Yes, with a 'managed fields' annotation.

Answer: B

Explanation:
RBAC in Kubernetesis coarse-grained: it controlsverbs(get, update, patch, delete) onresources(e.g., deployments), butnot individual fieldswithin a resource.
There isno /image subresource for deployments(there is one for pods but only for ephemeral containers).
Therefore,RBAC cannot restrict changes only to the image field.
Admission Webhooks(mutating/validating)canenforce fine-grained policies (e.g., deny updates that change anything other than spec.containers[*].image).
Exact extract (Kubernetes Docs – Admission Webhooks):
"Admission webhooks can be used to enforce custom policies on objects being admitted."
[References:, Kubernetes Docs — RBAC: https://kubernetes.io/docs/reference/access-authn-authz/rbac/, Kubernetes Docs — Admission Webhooks: https://kubernetes.io/docs/reference/access-authn-authz/extensible-admission-controllers/,]

NEW QUESTION 6
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 7
When using a cloud provider's managed Kubernetes service, who is responsible for maintaining the etcd cluster?

A. Kubernetes administrator
B. Namespace administrator
C. Cloud provider
D. Application developer

Answer: C

Explanation:
Inmanaged Kubernetes services(EKS, GKE, AKS), the control plane is operated by thecloud provider.
This includesetcd, API server, controller manager, scheduler.
Users manageworker nodes(in some models) and workloads, but not the control plane.
Exact extract (GKE Docs):
"The control plane, including the API server and etcd database, is managed and maintained by Google."
Similarly forEKSandAKS, etcd is fully managed by the provider.
[References:, GKE Architecture: https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture, EKS Architecture: https://docs.aws.amazon.com/eks/latest/userguide/eks-architecture.html, AKS Docs: https://learn.microsoft.com/en-us/azure/aks/concepts-clusters-workloads,]

NEW QUESTION 8
Which of the following statements correctly describes a container breakout?

A. A container breakout is the process of escaping the container and gaining access to the Pod's network traffic.
B. A container breakout is the process of escaping a container when it reaches its resource limits.
C. A container breakout is the process of escaping the container and gaining access to the cloud provider's infrastructure.
D. A container breakout is the process of escaping the container and gaining access to the host operating system.

Answer: D

Explanation:
Container breakoutrefers to an attacker escaping container isolation and reaching thehost OS.
Once the host is compromised, the attacker can accessother containers, Kubernetes nodes, or escalate further.
Exact extract (Kubernetes Security Docs):
??If an attacker gains access to a container, they may attempt a container breakout to gain access to the host system.??

Other options clarified:
A: Network access inside a Pod # breakout.
B: Resource exhaustion is aDoS, not a breakout.
C: Cloud infrastructure compromise is possibleafterhost compromise, but not the definition of breakout.
References:
Kubernetes Security Concepts: https://kubernetes.io/docs/concepts/security/
CNCF Security Whitepaper (Threats section):https://github.com/cncf/tag-security

NEW QUESTION 9
A Kubernetes cluster tenant can launch privileged Pods in contravention of therestricted Pod Security Standardmandated for cluster tenants and enforced by the built-inPodSecurity admission controller.
The tenant has full CRUD permissions on the namespace object and the namespaced resources. How did the tenant achieve this?

A. The scope of the tenant role means privilege escalation is impossible.
B. By tampering with the namespace labels.
C. By deleting the PodSecurity admission controller deployment running in their namespace.
D. By using higher-level access credentials obtained reading secrets from another namespace.

Answer: B

Explanation:
ThePodSecurity admission controllerenforces Pod Security Standards (Baseline, Restricted, Privileged)based on namespace labels.
If a tenant has full CRUD on the namespace object, they canmodify the namespace labelsto remove or weaken the restriction (e.g., setting pod-security.kubernetes.io/enforce=privileged).
This allows privileged Pods to be admitted despite the security policy.

Incorrect options:
(A) is false — namespace-level access allows tampering.
(C) is invalid — PodSecurity admission is not namespace-deployed, it??s a cluster-wide admission controller.
(D) is unrelated — Secrets from other namespaces wouldn??t directly bypass PodSecurity enforcement.
References:

Kubernetes Documentation – Pod Security Admission
CNCF Security Whitepaper – Admission control and namespace-level policy enforcement weaknesses.

NEW QUESTION 10
What mechanism can I use to block unsigned images from running in my cluster?

A. Enabling Admission Controllers to validate image signatures.
B. Using PodSecurityPolicy (PSP) to enforce image signing and validation.
C. Using Pod Security Standards (PSS) to enforce validation of signatures.
D. Configuring Container Runtime Interface (CRI) to enforce image signing and validation.

Answer: A

Explanation:
KubernetesAdmission Controllers(particularlyValidatingAdmissionWebhooks) can be used to enforce policies that validate image signatures.
This is commonly implemented withtools like Sigstore/cosign, Kyverno, or OPA Gatekeeper.
PodSecurityPolicy (PSP):deprecated and never supported image signature validation.
Pod Security Standards (PSS):only apply to pod security fields (privilege, users, host access), not image signatures.
CRI:while runtimes (containerd, CRI-O) may integrate with signature verification tools, enforcement in Kubernetes is generally done viaAdmission Controllersat the API layer.
Exact extract (Admission Controllers docs):
??Admission webhooks can be used to enforce custom policies on the objects being admitted.?? (e.g., validating signatures).
References:
Kubernetes Docs — Admission Controllers: https://kubernetes.io/docs/reference/access-authn-authz
/admission-controllers/
Sigstore Project (cosign): https://sigstore.dev/
Kyverno ImageVerify Policy: https://kyverno.io/policies/pod-security/require-image-verification/

NEW QUESTION 10
In which order are thevalidating and mutating admission controllersrun while the Kubernetes API server processes a request?

A. The order of execution varies and is determined by the cluster configuration.
B. Validating admission controllers run before mutating admission controllers.
C. Validating and mutating admission controllers run simultaneously.
D. Mutating admission controllers run before validating admission controllers.

Answer: D

Explanation:
Theadmission control flowin Kubernetes:
Mutating admission controllersrun first and can modify incoming requests.
Validating admission controllersrun after mutations to ensure the final object complies with policies.
This ensures policies validate thefinal, mutated object.
References:
Kubernetes Documentation – Admission Controllers CNCF Security Whitepaper – Admission control workflow.

NEW QUESTION 15
Why mightNetworkPolicyresources have no effect in a Kubernetes cluster?

A. NetworkPolicy resources are only enforced if the Kubernetes scheduler supports them.
B. NetworkPolicy resources are only enforced if the networking plugin supports them.
C. NetworkPolicy resources are only enforced for unprivileged Pods.
D. NetworkPolicy resources are only enforced if the user has the right RBAC permissions.

Answer: B

Explanation:
NetworkPolicies define how Pods can communicate with each other and external endpoints.
However, Kubernetes itselfdoes not enforce NetworkPolicy. Enforcement depends on theCNI plugin
used (e.g., Calico, Cilium, Kube-Router, Weave Net).
If a cluster is using a network plugin that does not support NetworkPolicies, then creating NetworkPolicy objects hasno effect.
References:
Kubernetes Documentation – Network Policies
CNCF Security Whitepaper – Platform security section: notes that security enforcement relies on CNI capabilities.

NEW QUESTION 16
How do Kubernetes namespaces impact the application of policies when using Pod Security Admission?

A. Namespaces are ignored; Pod Security Admission policies apply cluster-wide only.
B. Different policies can be applied to specific namespaces.
C. Each namespace can have only one active policy.
D. The default namespace enforces the strictest security policies by default.

Answer: B

Explanation:
Pod Security Admission (PSA)enforces policies by applyinglabels on namespaces, not globally across the cluster.
Exact extract (Kubernetes Docs – Pod Security Admission):
??You can apply Pod Security Standards to namespaces by adding labels such as pod- security.kubernetes.io/enforce. Different namespaces can enforce

different policies.??
Clarifications:
A: Incorrect, namespaces are the unit of enforcement.
C: Misleading — a namespace can have multiple enforcement modes (enforce, audit, warn).
D: Default namespace doesnotenforce strict policies unless labeled.
References:
Kubernetes Docs — Pod Security Admission: https://kubernetes.io/docs/concepts/security/pod-security- admission/

NEW QUESTION 18
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 23
You are responsible for securing thekubeletcomponent in a Kubernetes cluster.
Which of the following statements about kubelet security is correct?

A. Kubelet runs as a privileged container by default.
B. Kubelet does not have any built-in security features.
C. Kubelet supports TLS authentication and encryption for secure communication with the API server.
D. Kubelet requires root access to interact with the host system.

Answer: C

Explanation:
Thekubeletis the primary agent that runs on each node in a Kubernetes cluster and communicates with the control plane.
Kubeletsupports TLS (Transport Layer Security)for both authentication and encryption when interacting with the API server. This is a core security feature that ensures secure node-to-control-plane communication.
Incorrect options:
(A) Kubelet does not run as a privileged container by default; it runs as a system process (typically systemd-managed) on the host.
(B) Kubelet does include built-in security features such asTLS authentication, authorization modes, and read-only vs secured ports.
(D) While kubelet interacts with the host system (e.g., cgroups, container runtimes), it does not inherently require root access for communication security; RBAC and TLS handle authentication.
[References:, Kubernetes Documentation – Kubelet authentication/authorization, CNCF Security Whitepaper – Cluster Component Security (discusses TLS and mutual authentication between kubelet and API server).,]

NEW QUESTION 26
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 28
Which step would give an attacker a foothold in a cluster butno long-term persistence?

A. Modify Kubernetes objects stored within etcd.
B. Modify file on host filesystem.
C. Starting a process in a running container.
D. Create restarting container on host using Docker.

Answer: C

Explanation:
Starting a process in a running containerprovides an attacker withtemporary execution (foothold)inside the cluster, but once the container is stopped or restarted, that malicious process is lost. This means the attacker has nolong-term persistence.
Incorrect options:
(A) Modifying objects inetcdgrants persistent access since cluster state is stored in etcd.
(B) Modifying files on thehost filesystemcan create persistence across reboots or container restarts.
(D) Creating a restarting container directly on the host via Docker bypasses Kubernetes but persists across pod restarts if Docker restarts it.
[References:, CNCF Security Whitepaper – Threat Modeling section: Describes howephemeral processes inside containersprovide attackers short-term control but not durable persistence., Kubernetes Documentation – Cluster Threat Model emphasizes ephemeral vs. persistent attacker footholds.,]

NEW QUESTION 32
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 35
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 39
.....

Relate Links

100% Pass Your KCSA Exam with Exambible Prep Materials

https://www.exambible.com/KCSA-exam/

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - https://www.exambible.com/