Linux-Foundation

Exam Questions KCSA

Kubernetes and Cloud Native Security Associate (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
Which information does a user need to verify a signed container image?

A. The image's SHA-256 hash and the private key of the signing authority.
B. The image's digital signature and the private key of the signing authority.
C. The image's SHA-256 hash and the public key of the signing authority.
D. The image's digital signature and the public key of the signing authority.

Answer: D

Explanation:

Container image signing (e.g., withcosign, Notary v2) uses asymmetric cryptography.

Verification process:

Retrieve theimage??s digital signature.

Validate the signature with thepublic keyof the signer.

Exact extract (Sigstore Cosign Docs):

??Verification of an image requires the signature and the signer??s public key. The signature proves authenticity and integrity.??

Why others are wrong:

A & B: The private key is only used by the signer, never shared.

C: The hash alone cannot prove authenticity without the digital signature.

References:
Sigstore Cosign Docs: https://docs.sigstore.dev/cosign/overview

NEW QUESTION 4
Which other controllers are part of the kube-controller-manager inside the Kubernetes cluster?

A. Job controller, CronJob controller, and DaemonSet controller
B. Pod, Service, and Ingress controller
C. Namespace controller, ConfigMap controller, and Secret controller
D. Replication controller, Endpoints controller, Namespace controller, and ServiceAccounts controller

Answer: D

Explanation:

kube-controller-managerruns a set of controllers that regulate the cluster??s state.

Exact extract (Kubernetes Docs): "The kube-controller-manager runs controllers that are core to Kubernetes. Examples of controllers are: Node controller, Replication controller, Endpoints controller, Namespace controller, and ServiceAccounts controller."

Why D is correct:All listed are actual controllers within kube-controller-manager.

Why others are wrong:

A:Job and CronJob controllers are managed by kube-controller-manager, but DaemonSet controller is managed by the kube-scheduler/deployment logic.

B:Pod, Service, Ingress controllers are not part of kube-controller-manager.

C:ConfigMap and Secret do not have dedicated controllers.
[References:, Kubernetes Docs — kube-controller-manager: https://kubernetes.io/docs/reference/command-line-tools-reference/kube-controller-manager/,]

NEW QUESTION 5
A cluster administrator wants to enforce the use of a different container runtime depending on the application a workload belongs to.

A. By manually modifying the container runtime for each workload after it has been created.
B. By modifying the kube-apiserver configuration file to specify the desired container runtime for each application.
C. By configuring avalidating admission controllerwebhook that verifies the container runtime based on the application label and rejects requests that do not comply.
D. By configuring amutating admission controllerwebhook that intercepts new workload creation requests and modifies the container runtime based on the application label.

Answer: D

Explanation:

Kubernetes supports workload-specific runtimes viaRuntimeClass.

Amutating admission controllercan enforce this automatically by:

Intercepting workload creation requests.

Modifying the Pod spec to set runtimeClassName based on labels or policies.

Incorrect options:
(A) Manual modification is not scalable or secure.
(B) kube-apiserver cannot enforce per-application runtime policies.
(C) A validating webhook can onlyreject, not modify, the runtime.
[References:, Kubernetes Documentation – RuntimeClass, CNCF Security Whitepaper – Admission controllers for enforcing runtime policies.,]

NEW QUESTION 6
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 7
Which technology can be used to apply security policy for internal cluster traffic at the application layer of the network?

A. Network Policy
B. Ingress Controller
C. Container Runtime
D. Service Mesh

Answer: D

Explanation:

Service Mesh (e.g., Istio, Linkerd, Consul):operates atLayer 7 (application layer), enforcing policies like mTLS, authorization, and routing between services.

NetworkPolicy:works atLayer 3/4 (IP/port), not Layer 7.

Ingress Controller:handles external traffic ingress, not internal service-to-service traffic.

Container Runtime:responsible for running containers, not enforcing application-layer security.

Exact extract (Istio docs):
"Istio provides security by enforcing authentication, authorization, and encryption of service-to-service communication."
[References:, Kubernetes Docs — Network Policies: https://kubernetes.io/docs/concepts/services-networking/network-policies/, Istio Security Docs: https://istio.io/latest/docs/concepts/security/,]

NEW QUESTION 8
What was the name of the precursor to Pod Security Standards?

A. Container Runtime Security
B. Kubernetes Security Context
C. Container Security Standards
D. Pod Security Policy

Answer: D

Explanation:
Kubernetes originally had a feature calledPodSecurityPolicy (PSP), which provided controls to restrict pod behavior.
Official docs:
"PodSecurityPolicy was deprecated in Kubernetes v1.21 and removed in v1.25."
"Pod Security Standards (PSS) replace PodSecurityPolicy (PSP) with a simpler, policy-driven approach."
PSP was often complex and hard to manage, so it was replaced by Pod Security Admission (PSA) which enforcesPod Security Standards.
[References:, Kubernetes Docs — PodSecurityPolicy (deprecated): https://kubernetes.io/docs/concepts/security/pod-security-policy/, Kubernetes Blog — PodSecurityPolicy Deprecation: https://kubernetes.io/blog/2021/04/06/podsecuritypolicy-deprecation-past-present-and-future/,]

NEW QUESTION 9
An attacker has access to the network segment that the cluster is on.
What happens when a compromised Pod attempts to connect to the API server?

A. The compromised Pod is automatically isolated from the network to prevent any connections to the API server.
B. The compromised Pod is allowed to connect to the API server without any restrictions.
C. The compromised Pod attempts to connect to the API server, but its requests may be blocked due to network policies.
D. The compromised Pod connects to the API server and is granted elevated privileges by default.

Answer: C

Explanation:
By default,Pods can connect to the API server(since ServiceAccount tokens are mounted).
However, whether they succeed in acting depends on:
Network Policies(may block egress).
RBAC(controls permissions).

Exact extract (Kubernetes Docs – API Access):
??Pods authenticate to the API server using the service account token mounted into the Pod.
Authorization is then enforced by RBAC. NetworkPolicies may further restrict access.??

Clarifications:
A: No default automatic isolation.
B: Not always unrestricted; policies may apply.
D: Pods get minimal default privileges, not automatic elevation.
References:
Kubernetes Docs — API Access to Pods: https://kubernetes.io/docs/concepts/security/service-accounts/ Kubernetes Docs — Network Policies: https://kubernetes.io/docs/concepts/services-networking/network-policies/

NEW QUESTION 10
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 10
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 12
Which way of defining security policy brings consistency, minimizes toil, and reduces the probability of misconfiguration?

A. Using a declarative approach to define security policies as code.
B. Relying on manual audits and inspections for security policy enforcement.
C. Manually configuring security controls for each individual resource, regularly.
D. Implementing security policies through manual scripting on an ad-hoc basis.

Answer: A

Explanation:
Defining policiesas code (declarative)is a best practice in Kubernetes and cloud-native security.
This is aligned withGitOpsandPolicy-as-Codeprinciples (OPA Gatekeeper, Kyverno, etc.).
Exact extract (CNCF Security Whitepaper):
??Policy-as-Code enables declarative definition and enforcement of security policies, bringing consistency, automation, and reducing misconfiguration risk.??
Manual audits, ad-hoc scripting, or individual configurations are error-prone and inconsistent.
References:

CNCF Security Whitepaper:https://github.com/cncf/tag-security
Kubernetes Docs — Policy as Code (OPA, Kyverno): https://kubernetes.io/docs/concepts/security/

NEW QUESTION 13
As a Kubernetes and Cloud Native Security Associate, a user can set upaudit loggingin a cluster. What is the risk of logging every event at the fullRequestResponselevel?

A. No risk, as it provides the most comprehensive audit trail.
B. Increased storage requirements and potential impact on performance.
C. Improved security and easier incident investigation.
D. Reduced storage requirements and faster performance.

Answer: B

Explanation:
Audit loggingrecords API server requests and responses for security monitoring.
TheRequestResponse levellogs the full request and response bodies, which can:
Significantly increasestorage and performance overhead.
Potentially log sensitive data (including Secrets).
Therefore, while comprehensive, it introduces risks of performance degradation and excessive log volume.
References:
Kubernetes Documentation – Auditing
CNCF Security Whitepaper – Logging and monitoring: trade-offs between verbosity, storage, and security.

NEW QUESTION 14
When should soft multitenancy be used over hard multitenancy?

A. When the priority is enabling resource sharing and efficiency between tenants.
B. When the priority is enabling complete isolation between tenants.
C. When the priority is enabling fine-grained control over tenant resources.
D. When the priority is enabling strict security boundaries between tenants.

Answer: A

Explanation:
Soft multitenancy(Namespaces, RBAC, Network Policies) # assumes some level of trust between tenants, focuses onresource sharing and efficiency.
Hard multitenancy(separate clusters or strong virtualization) # strict isolation, used when tenants are untrusted.
Exact extract (CNCF TAG Security Multi-Tenancy Whitepaper):
??Soft multi-tenancy refers to multiple workloads running in the same cluster with some trust assumptions. It provides resource sharing and operational efficiency.
Hard multi- tenancy requires stronger isolation guarantees, typically separate clusters.??
References:
CNCF Security TAG — Multi-Tenancy Whitepaper:https://github.com/cncf/tag-security/tree/main/multi- tenancy

NEW QUESTION 19
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 20
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 22
What is the reasoning behind considering the Cloud as the trusted computing base of a Kubernetes cluster?

A. The Cloud enforces security controls at the Kubernetes cluster level, so application developers can focus on applications only.
B. A Kubernetes cluster can only be trusted if the underlying Cloud provider is certified against international standards.
C. A vulnerability in the Cloud layer has a negligible impact on containers due to Linux isolation mechanisms.
D. A Kubernetes cluster can only be as secure as the security posture of its Cloud hosting.

Answer: D

Explanation:
The4C??s of Cloud Native Security(Cloud, Cluster, Container, Code) model starts withCloudas the base layer.
If the Cloud (infrastructure layer) is compromised, every higher layer (Cluster, Container, Code) inherits that compromise.
Exact extract (Kubernetes Security Overview):
??The 4C??s of Cloud Native security are Cloud, Clusters, Containers, and Code. You can think of the 4C??s as a layered approach. A Kubernetes cluster can only be as secure as the cloud infrastructure it is deployed on.??
This means the cloud is part of thetrusted computing baseof a Kubernetes cluster.
References:
Kubernetes Docs — Security Overview (4C??s): https://kubernetes.io/docs/concepts/security/overview/#the- 4cs-of-cloud-native-security

NEW QUESTION 24
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 27
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 29
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 34
To restrict the kubelet's rights to the Kubernetes API, whatauthorization modeshould be set on the Kubernetes API server?

A. Node
B. AlwaysAllow
C. kubelet
D. Webhook

Answer: A

Explanation:
TheNode authorization modeis designed to specifically limit what kubelets can do when they connect to the Kubernetes API server.
It authorizes requests from kubelets based on the Pods scheduled to run on their nodes, ensuring kubelets cannot interact with resources beyond their scope.
Incorrect options:
(B)AlwaysAllowallows unrestricted access (insecure).
(C) No kubelet authorization mode exists.
(D)Webhookmode delegates authorization decisions to an external service, not specifically for kubelets.
[References:, Kubernetes Documentation – Node Authorization, CNCF Security Whitepaper – Access control: kubelet authorization and Node authorizer.,]

NEW QUESTION 39
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 43
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 48
What is Grafana?

A. A cloud-native distributed tracing system for monitoring microservices architectures.

B. A container orchestration platform for managing and scaling applications.
C. A platform for monitoring and visualizing time-series data.
D. A cloud-native security tool for scanning and detecting vulnerabilities in Kubernetes clusters.

Answer: C

Explanation:
Grafana:An open-source analytics and visualization platform widely used with Prometheus, Loki, etc.
Exact extract (Grafana Docs):??Grafana is the open-source analytics and monitoring solution for every database. It allows you to query, visualize, alert on, and understand your metrics no matter where they are stored.??
A is wrong:That describesJaeger(distributed tracing).
B is wrong:That??sKubernetesitself.
D is wrong:That??sTrivy/Aqua/Prismatype tools.
References:
Grafana Docs: https://grafana.com/docs/grafana/latest/

NEW QUESTION 52
What is the purpose of the Supplier Assessments and Reviews control in the NIST 800-53 Rev. 5 set of controls for Supply Chain Risk Management?

A. To evaluate and monitor existing suppliers for adherence to security requirements.
B. To conduct regular audits of suppliers' financial performance.
C. To establish contractual agreements with suppliers.
D. To identify potential suppliers for the organization.

Answer: A

Explanation:
In NIST SP 800-53 Rev. 5,SR-6: Supplier Assessments and Reviewsrequires evaluating and monitoring suppliers' security and risk practices.
Exact extract (NIST SP 800-53 Rev. 5, SR-6):
"The organization assesses and monitors suppliers to ensure they are meeting the security requirements specified in contracts and agreements."
This is aboutongoing monitoringof supplier adherence, not financial audits, not contract creation, and not supplier discovery.
References:
NIST SP 800-53 Rev. 5, Control SR-6 (Supplier Assessments and Reviews): https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final

NEW QUESTION 56
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 58
.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

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