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
What information is stored in etcd?

A. Etcd manages the configuration data, state data, and metadata for Kubernetes.
B. Application logs and monitoring data for auditing and troubleshooting purposes.
C. Sensitive user data such as usernames and passwords.
D. Pod data contained in Persistent Volume Claims (e.
E. hostPath).

Answer: A

Explanation:

- etcdis Kubernetes??key-value storeforcluster state.
- Stores: ConfigMaps, Secrets, Pod definitions, Deployments, RBAC policies, and metadata.
- Exact extract (Kubernetes Docs – etcd):
- ??etcd is a consistent and highly-available key-value store used as Kubernetes?? backing store for all cluster data.??
- Clarifications:
- B: Logs/metrics are handled by logging/monitoring solutions, not etcd.
- C: Secrets may be stored here but encoded in base64, not specifically "usernames/passwords" as primary use.
- D: Persistent Volumes are external storage, not stored in etcd.

References:
Kubernetes Docs — etcd: https://kubernetes.io/docs/concepts/overview/components/#etcd

NEW QUESTION 2
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 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
In Kubernetes, what isPublic Key Infrastructure (PKI)used for?

A. To manage certificates and ensure secure communication in a Kubernetes cluster.
B. To automate the scaling of containers in a Kubernetes cluster.
C. To manage networking in a Kubernetes cluster.
D. To monitor and analyze performance metrics of a Kubernetes cluster.

Answer: A

Explanation:
Kubernetes usesPKI certificatesextensively to secure communication between control plane components (API server, etcd, kube-scheduler, kube-controller-manager) and with kubelets.
Certificates enablemutual TLS authentication and encryptionacross components.
PKI does not handle scaling, networking, or monitoring.
References:
Kubernetes Documentation – Certificates
CNCF Security Whitepaper – Cluster communication security and the role of PKI.

NEW QUESTION 5
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 6
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 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
In the event that kube-proxy is in a CrashLoopBackOff state, what impact does it have on the Pods running on the same worker node?

A. The Pods cannot communicate with other Pods in the cluster.
B. The Pod cannot mount persistent volumes through CSI drivers.
C. The Pod's security context restrictions cannot be enforced.
D. The Pod's resource utilization increases significantly.

Answer: A

Explanation:
kube-proxy:manages cluster network routing rules (via iptables or IPVS). It enables Pods to communicate with Services and Pods across nodes.
If kube-proxy fails (CrashLoopBackOff), service IP routing and cluster-wide pod-to-pod networking breaks. Local Pod-to-Pod communication within the same node may still work, butcross-node communication fails.
Exact extract (Kubernetes Docs – kube-proxy):
"kube-proxy maintains network rules on nodes. These rules allow network communication to Pods from network sessions inside or outside of the cluster."
[References:, Kubernetes Docs — kube-proxy: https://kubernetes.io/docs/reference/command-line-tools-reference/kube-proxy/,]

NEW QUESTION 9
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 10
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 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
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 17
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 20
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 23
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 25
Which label should be added to the Namespace to block any privileged Pods from being created in that Namespace?

A. privileged: false
B. privileged: true
C. pod-security.kubernetes.io/enforce: baseline
D. pod.security.kubernetes.io/privileged: false

Answer: C

Explanation:
KubernetesPod Security Admission (PSA)enforcesPod Security Standardsby applying labels on Namespaces.
Exact extract (Kubernetes Docs – Pod Security Admission):
??You can label a namespace with pod-security.kubernetes.io/enforce: baseline to enforce the Baseline policy.??
Thebaselineprofile explicitly disallowsprivileged podsand other unsafe features.
Why others are wrong:
A & D: These labels do not exist in Kubernetes.
B: Setting privileged: true would allow privileged pods, not block them.
References:
Kubernetes Docs — Pod Security Admission: https://kubernetes.io/docs/concepts/security/pod-security- admission/
Kubernetes Docs — Pod Security Standards: https://kubernetes.io/docs/concepts/security/pod-security- standards/

NEW QUESTION 30
An attacker compromises a Pod and attempts to use its service account token to escalate privileges within the cluster. Which Kubernetes security feature is designed tolimit what this service account can do?

A. PodSecurity admission
B. NetworkPolicy
C. Role-Based Access Control (RBAC)
D. RuntimeClass

Answer: C

Explanation:
When a Pod is created, Kubernetes automatically mounts aservice account tokenthat can authenticate to the API server.
TheRole-Based Access Control (RBAC)system defines what actions a service account can perform.
By carefully restricting Roles and RoleBindings, administrators limit the blast radius of a compromised
Pod.
Incorrect options:
(A)PodSecurity admissionenforces workload-level security settings but does not control API access.
(B)NetworkPolicycontrols network communication, not API privileges.
(D)RuntimeClassselects container runtimes, unrelated to privilege escalation through API tokens.
References:
Kubernetes Documentation – Using RBAC Authorization
CNCF Security Whitepaper – Identity & Access Management: limiting lateral movement by constraining service account permissions.

NEW QUESTION 35
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 37
Which security knowledge-base focuses specifically onoffensive tools, techniques, and procedures?

A. MITRE ATT&CK
B. OWASP Top 10
C. CIS Controls
D. NIST Cybersecurity Framework

Answer: A

Explanation:

MITRE ATT&CKis a globally recognizedknowledge base of adversary tactics, techniques, and procedures (TTPs). It is focused on describingoffensive behaviorsattackers use.
Incorrect options:
(B)OWASP Top 10highlights common application vulnerabilities, not attacker techniques.
(C)CIS Controlsare defensive best practices, not offensive tools.
(D)NIST Cybersecurity Frameworkprovides a risk-based defensive framework, not adversary TTPs.
References:
MITRE ATT&CK Framework
CNCF Security Whitepaper – Threat intelligence section: references MITRE ATT&CK for describing attacker behavior.

NEW QUESTION 38
What is the purpose of an egress NetworkPolicy?

A. To control the incoming network traffic to a Kubernetes cluster.
B. To control the outbound network traffic from a Kubernetes cluster.
C. To secure the Kubernetes cluster against unauthorized access.
D. To control the outgoing network traffic from one or more Kubernetes Pods.

Answer: D

Explanation:
NetworkPolicycontrols network trafficat the Pod level.
Ingress rules:controlincomingconnections to Pods.
Egress rules:controloutgoingconnectionsfrom Pods.
Exact extract (Kubernetes Docs – Network Policies):
"An egress rule controls outgoing connections from Pods that match the policy."
Clarifying wrong answers:
A/B: Too broad (cluster-level); policies apply per Pod/Namespace.
C: Security against unauthorized access is broader than egress policies.
[References:, Kubernetes Docs — Network Policies: https://kubernetes.io/docs/concepts/services-networking/network-policies/,]

NEW QUESTION 40
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 41
In order to reduce the attack surface of the Scheduler, which default parameter should be set to false?

A. --scheduler-name
B. --profiling
C. --secure-kubeconfig
D. --bind-address

Answer: B

Explanation:
Thekube-schedulerexposes aprofiling/debugging endpointwhen --profiling=true (default).
This can unnecessarily increase the attack surface.
Best practice: set --profiling=false in production.
Exact extract (Kubernetes Docs – kube-scheduler flags):
"--profiling (default true): Enable profiling via web interface host:port/debug/pprof/."
Why others are wrong:
--scheduler-name: just identifies the scheduler, not a security risk.
--secure-kubeconfig: not a valid flag.
--bind-address: changing it limits exposure but is not the default risk parameter for profiling.
[References:, Kubernetes Docs — kube-scheduler options: https://kubernetes.io/docs/reference/command-line-tools-reference/kube-scheduler/, ,]

NEW QUESTION 45
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 50
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 51
.....

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/