# SECRET-SEN$^{\text{Q\&As}}$

CyberArk Sentry - Secrets Manager

## Pass CyberArk SECRET-SEN Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

**https://www.geekcert.com/secret-sen.html**

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by CyberArk Official Exam Center

**QUESTION 1**

What is the most maintenance-free way to ensure a Conjur host\\'s access reflects any changes made to accounts in a safe in the CyberArk vault?

A. Write an automation script to update and load the host\\'s policy using PATCH/update.

B. Use yami anchor [and] and wildcard (*) syntax to maintain its list of permission grants.

C. Grant the consumers group/role created by the Synchronizer for the Safe to the host.

D. Use PVWA to add the Conjur host ID as a member of the Safe.

Correct Answer: C

The most maintenance-free way to ensure a Conjur host\\'s access reflects any changes made to accounts in a safe in the CyberArk vault is to grant the consumers group/role created by the Synchronizer for the Safe to the host. This means

that the host will inherit the read and execute permissions on all the secrets in the Safe from the consumers group/role, and will automatically get access to any new or updated secrets in the Safe without requiring any manual intervention or

policy changes. The consumers group/role is created by the Vault Conjur Synchronizer, which is a service that synchronizes secrets between the CyberArk vault and Conjur. The Synchronizer creates a policy branch for each Safe in Conjur,

and assigns the consumers group/role to have read and execute permissions on all the secrets in the Safe. The Synchronizer also creates a delegation policy for each Safe, which allows the Safe admins to grant permissions to other users,

hosts, groups, or layers12.

The other options are not the most maintenance-free ways to ensure a Conjur host\\'s access reflects any changes made to accounts in a safe in the CyberArk vault. Writing an automation script to update and load the host\\'s policy using

PATCH/update may work, but it requires additional effort and maintenance to ensure the script is always running and up to date with the changes in the Safe. Using yami anchor [and] and wildcard (*) syntax to maintain its list of permission

grants may simplify the policy writing, but it still requires manual editing and loading of the policy whenever a new secret is added or removed from the Safe. Using PVWA to add the Conjur host ID as a member of the Safe may not be

possible or advisable, as the PVWA is designed for managing human users and not Conjur hosts, and it may not have the necessary integration or authorization to do so3.

References:

Vault Conjur Synchronizer 1, Synchronizer Policy Structure Grant permissions on secrets 2, Grant role permissions on all secrets in a Safe Privileged Access Manager - Self-Hosted 3, Privileged Web Access (PVWA)

**QUESTION 2**

When an application is retrieving a credential from Conjur, the application authenticates to Follower A. Follower B receives the next request to retrieve the credential.

What happens next?

A. The Coniur Token is stateless and Follower B is able to validate the Token and satisfy the request.

B. The Coniur Token is stateful and Follower B is unable to validate the Token promptinq the application to re-authenticate.

C. The Coryur Token is stateless and Follower B redirects the request to Follower A to satisfy the request.

D. The Coniur Token is stateful and Follower B redirects the request to Follower A to satisfy the request.

Correct Answer: A

This is the correct answer because the Conjur Token is a JSON Web Token (JWT) that is signed by the Conjur master and contains the identity and permissions of the application. The Conjur Token is stateless, meaning that it does not depend on any stored session or transaction information on the server side. Therefore, any Conjur follower can validate the Token by verifying the signature and the expiration time, and satisfy the request by retrieving the credential from the local database. This allows the Conjur followers to be horizontally scalable and load balanced, and to provide high availability and performance for the applications. This answer is based on the Conjur documentation1 and the Conjur training course2.

**QUESTION 3**

You are setting up a Kubernetes integration with Conjur. With performance as the key deciding factor, namespace and service account will be used as identity characteristics.

Which authentication method should you choose?

A. JWT-based authentication

B. Certificate-based authentication

C. API key authentication

D. Connect (OIDC) authentication

Correct Answer: A

According to the CyberArk Sentry Secrets Manager documentation, JWT- based authentication is the recommended method for authenticating Kubernetes pods with Conjur. JWT-based authentication uses JSON Web Tokens (JWTs) that are issued by the Kubernetes API server and signed by its private key. The JWTs contain the pod\\'s namespace and service account as identity characteristics, which are verified by Conjur against a policy that defines the allowed namespaces and service accounts. JWT-based authentication is fast, scalable, and secure, as it does not require any additional certificates, secrets, or sidecars to be deployed on the pods. JWT-based authentication also supports rotation and revocation of the Kubernetes API server\\'s private key, which enhances the security and resilience of the authentication process. Certificate-based authentication is another method for authenticating Kubernetes pods with Conjur, but it is not the best option for performance. Certificate-based authentication uses X.509 certificates that are generated by a Conjur CA service and injected into the pods as Kubernetes secrets. The certificates contain the pod\\'s namespace and service account as identity characteristics, which are verified by Conjur against a policy that defines the allowed namespaces and service accounts. Certificate-based authentication is secure and reliable, but it requires more resources and steps to generate, inject, and manage the certificates and secrets. Certificate-based authentication also does not support rotation and revocation of the certificates, which may pose a security risk if the certificates are compromised or expired. API key authentication and Connect (OIDC) authentication are not valid methods for authenticating Kubernetes pods with Conjur. API key authentication is used for authenticating hosts, users, and applications that have a Conjur identity and an API key. Connect (OIDC) authentication is used for authenticating users and applications that have an OpenID Connect identity and a token. These methods are not suitable for Kubernetes pods, as they do not use the pod\\'s namespace and service account as identity characteristics, and they require additional secrets or tokens to be stored and managed on the pods. References: = JWT Authenticator | CyberArk Docs; Certificate Authenticator | CyberArk Docs; API Key Authenticator | CyberArk Docs; Connect Authenticator | CyberArk Docs

**QUESTION 4**

A customer has 100 .NET applications and wants to use Summon to invoke the application and inject secrets at run time.

Which change to the NET application code might be necessary to enable this?

A. It must be changed to include the REST API calls necessary to retrieve the needed secrets from the CCP.

B. It must be changed to access secrets from a configuration file or environment variable.

C. No changes are needed as Summon brokers the connection between the application and the backend data source through impersonation.

D. It must be changed to include the host API key necessary for Summon to retrieve the needed secrets from a Follower

Correct Answer: B

Summon is a utility that allows applications to access secrets from a variety of trusted stores and export them as environment variables to a sub-process environment. Summon does not require any changes to the application code to retrieve secrets from the CyberArk Central Credential Provider (CCP), as it uses a provider plugin that handles the communication with the CCP. However, the application code must be able to access secrets from a configuration file or environment variable, as these are the methods that Summon uses to inject secrets into the application. Summon reads a secrets.yml file that defines the secrets that the application needs and maps them to environment variables. Then, Summon fetches the secrets from the CCP using the provider plugin and exports them as environment variables to the application sub-process. The application can then read the secrets from the environment variables as if they were hard-coded in the configuration file. References: Summon-inject secrets, .NET Application Password SDK

**QUESTION 5**

You have a request to protect all the properties around a credential object. When configuring the credential in the Vault, you specified the address, user and password for the credential.

How do you configure the Vault Conjur Synchronizer to properly sync all properties?

A. Modify VaultConjurSynchronizer.exe.config, uncomment SYNCALLPROPERTIES and update its value to true.

B. Modify SynchronizerReplication.config, uncomment SYNCALLPROPERTIES and update its value to true.

C. Modify Vault.ini, uncomment SYNCALLPROPERTIES and update its value to true.

D. In the Conjur UI under Cluster > Synchronizer > Config, change SYNCALLPROPERTIES and update its value to true.

Correct Answer: B

This is the correct answer because the SynchronizerReplication.config file contains the configuration settings for the Vault Conjur Synchronizer service (Synchronizer) to sync secrets from the CyberArk Vault to the Conjur database. The SYNCALLPROPERTIES parameter specifies whether to sync all the properties of the accounts in the Vault or only the password property. By default, the SYNCALLPROPERTIES parameter is set to false, which means that only the password property is synced. To sync all the properties, such as the address and the user, the SYNCALLPROPERTIES parameter needs to be set to true. This answer is based on the CyberArk Secrets Manager documentation1 and the CyberArk Secrets Manager training course2. The other options are not correct because they do not configure the Synchronizer to properly sync all properties. Modifying VaultConjurSynchronizer.exe.config, uncommenting SYNCALLPROPERTIES and updating its value to true is not a valid option, as this file does not contain the SYNCALLPROPERTIES parameter. The VaultConjurSynchronizer.exe.config file contains the configuration settings for the Synchronizer service, such as the log level, the log path, and the service name. The SYNCALLPROPERTIES parameter is only found in the SynchronizerReplication.config file. Modifying Vault.ini, uncommenting SYNCALLPROPERTIES and updating its value to true is not a valid option, as this file does not contain the SYNCALLPROPERTIES parameter. The Vault.ini file contains the configuration settings for the CyberArk Central Credential Provider (CCP) to connect to the Vault server and provide credentials to the applications. The SYNCALLPROPERTIES parameter is not related to the CCP configuration or functionality. In the Conjur UI under Cluster > Synchronizer > Config, changing SYNCALLPROPERTIES and updating its value to true is not a valid option, as this section does not exist in the Conjur UI. The Conjur UI does not have a Cluster, Synchronizer, or Config section. The Conjur UI has a Cluster Config section under Settings, but this section is used to configure the Conjur cluster settings, such as the master IP address, the follower IP address, and the seed fetcher IP address. The SYNCALLPROPERTIES parameter is not related to the Conjur cluster configuration or functionality.

SECRET-SEN Practice Test | SECRET-SEN Exam Questions | SECRET-SEN Braindumps