SECRET-SEN 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

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

You are diagnosing this log entry: From Conjur logs:

```
USERNAME_MISSING failed to authenticate with authenticator authn-jwt service team-
a:webservice:conjur/
authn-jwt/jenkins-test: CONJ00087E Failed to fetch JWKS from
'https://jenkins.tst.acme.com/jwtauth/conjur
jwk-set'. Reason: '#<OpenSSL::SSL::SSLError:SSL_connect returned=1 errno=o state=error:
certificate verify
failed (unable to get local issuer certificate)>'

Apr 25, 2022 11:35:06 AM FINE org.conjur.jenkins.jwauth.impl.JwtToken sign
Signing Token
Apr 25, 2022 11:35:07 AM FINE org.conjur.jenkins.api.ConjurAPI getAuthorizationToken
Authenticating with Conjur (JWT) authnPath=authn-jwt/jenkins-test
Apr 25, 2022 11:35:08 AM FINEST org.conjur.jenkins.api.ConjurAPI getAuthorizationToken
Conjur Authenticate response 401 – Unauthorized
Apr 25, 2022 11:35:08 AM FINE org.conjur.jenkins.credentials.CredentialsSupplier get
EXCEPTION: CredentialSuplier => Error authenticating to Conjur [401 – Unauthorized
```

Given these errors, which problem is causing the breakdown?

A. The Jenkins certificate chain is not trusted by Conjur.

B. The Conjur certificate chain is not trusted by Jenkins.

C. The JWT sent by Jenkins does not match the Conjur host annotations.

D. The Jenkins certificate is malformed and will not be trusted by Conjur.

Correct Answer: A

The log entry shows a failed authentication attempt with Conjur using the authn-jwt method. This method allows applications to authenticate with Conjur using JSON Web Tokens (JWTs) that are signed by a trusted identity provider. In this case, the application is Jenkins, which is a CI/CD tool that can integrate with Conjur using the Conjur Jenkins plugin. The plugin allows Jenkins to securely retrieve secrets from Conjur and inject them as environment variables into Jenkins pipelines or projects. The log entry indicates that the JWT sent by Jenkins was rejected by Conjur because of an SSL connection error. The error message says that the certificate chain of Jenkins could not be verified by Conjur, and that the certificate authority (CA) that signed the Jenkins certificate was unknown to Conjur. This means that the Jenkins certificate chain is not trusted by Conjur, and that Conjur does not have the CA certificate of Jenkins in its trust store. Therefore, Conjur cannot establish a secure and trusted connection with Jenkins, and cannot validate the JWT signature. To fix this problem, the Jenkins certificate chain needs to be trusted by Conjur. This can be done by copying the CA certificate of Jenkins to the Conjur server, and adding it to the Conjur trust store. The Conjur trust store is a directory that contains the CA certificates of the trusted identity providers for the authn-jwt method. The Conjur server also needs to be restarted for the changes to take effect. References: Conjur Jenkins Plugin; Conjur JWT Authentication; Conjur Trust Store

QUESTION 2

During the configuration of Conjur, what is a possible deployment scenario?

A. The Leader and Followers are deployed outside of a Kubernetes environment; Slandbys can run inside a Kubernetes environment.

B. The Conjur Leader cluster is deployed outside of a Kubernetes environment; Followers can run inside or outside the environment.

C. The Leader cluster is deployed outside a Kubernetes environment; Followers and Standbys can run inside or outside the environment.

D. The Conjur Leader cluster and Followers are deployed inside a Kubernetes environment.

Correct Answer: C

Conjur is a secrets management solution that securely stores and manages secrets and credentials used by applications, DevOps tools, and other systems. Conjur can be deployed in different scenarios, depending on the needs and preferences of the organization. One of the possible deployment scenarios is to deploy the Leader cluster outside a Kubernetes environment, and the Followers and Standbys inside or outside the environment. The Leader cluster is the primary node that handles all write operations and coordinates the replication of data to the Follower and Standby nodes. The Leader cluster consists of one active Leader node and one or more Standby nodes that can be promoted to Leader in case of a failure. The Leader cluster can be deployed outside a Kubernetes environment, such as on a virtual machine or a physical server, using Docker or other installation methods. This can provide more control and flexibility over the configuration and management of the Leader cluster, as well as better performance and security. The Follower and Standby nodes are read-only replicas of the Leader node that can serve requests from clients and applications that need to retrieve secrets or perform other read- only operations. The Follower and Standby nodes can be deployed inside or outside a Kubernetes environment, depending on the use case and the availability requirements. For example, if the clients and applications are running inside a Kubernetes cluster, it may be convenient and efficient to deploy the Follower and Standby nodes inside the same cluster, using Helm charts or other methods. This can reduce the network latency and complexity, and leverage the Kubernetes features such as service discovery, load balancing, and health checks. Alternatively, if the clients and applications are running outside a Kubernetes cluster, or if there is a need to distribute the Follower and Standby nodes across different regions or availability zones, it may be preferable to deploy the Follower and Standby nodes outside the Kubernetes cluster, using Docker or other methods. This can provide more scalability and resiliency, and avoid the dependency on the Kubernetes cluster. References: Conjur Deployment Scenarios; Conjur Cluster Installation; Conjur Kubernetes Integration

QUESTION 3

In a 3-node auto-failover cluster, the Leader has been brought down for patching that lasts longer than the configured TTL. A Standby has been promoted.

Which steps are required to repair the cluster when the old Leader is brought back online?

A. On the new Leader, generate a Standby seed for the old Leader node and add it to the cluster member list. Rebuild the old Leader as a new Standby and then re-enroll the node to the cluster.

B. Generate a Standby seed for the newly promoted Leader. Stop and remove the container on the new Leader, then rebuild it as a new Standby. Re-enroll the Standby to the cluster and re-base replication of the 3rd Standby back to the old Leader.

C. Generate standby seeds for the newly-promoted Leader and the 3rd Standby Stop and remove the containers and then rebuild them as new Standbys. On both new Standbys, re-enroll the node to the cluster.

D. On the new Leader, generate a Standby seed for the old Leader node and re-upload the auto-failover policy in "replace" mode. Rebuild the old Leader as a new Standby, then re-enroll the node to the cluster.

Correct Answer: A

The correct answer is A. On the new Leader, generate a Standby seed for the old Leader node and add it to the cluster member list. Rebuild the old Leader as a new Standby and then re-enroll the node to the cluster. This is the recommended way to repair the cluster health after an auto-failover event, according to the CyberArk Sentry Secrets Manager documentation1. This method reuses the original Leader as a new Standby, without affecting the new Leader or the other Standby. The steps are as follows: On the new Leader, generate a Standby seed for the old Leader node using the command evoke seed standby . This will create a file named .tar in the current directory. On the new Leader, add the old Leader node to the cluster member list using the command evoke cluster add . On the old Leader server, stop and remove the container using the commands docker stop and docker rm . On the old Leader server, copy the Standby seed file from the new Leader using the command scp :.tar . On the old Leader server, create a new container using the same name as the one you just destroyed, and load the Standby seed file using the command docker run --name -d --restart=always -v /var/log/conjur:/var/log/conjur -v /opt/conjur/backup:/opt/conjur/backup -p "443:443" -p "5432:5432" -p "1999:1999" cyberark/conjur:latest seed fetch .tar On the old Leader server, re-enroll the node to the cluster using the command evoke cluster enroll The other options are not correct, as they either involve unnecessary or harmful steps, such as rebuilding the new Leader or the other Standby, or re-uploading the auto-failover policy in replace mode, which may cause data loss or inconsistency.

QUESTION 4

While retrieving a secret through REST, the secret retrieval fails to find a matching secret. You know the secret onboarding process was completed, the secret is in the expected safe with the expected object name, and the CCP is able to provide secrets to other applications.

What is the most likely cause for this issue?

A. The application ID or Application Provider does not have the correct permissions on the safe.

B. The client certificate fingerprint is not trusted.

C. The service account running the application does not have the correct permissions on the safe.

D. The OS user does not have the correct permissions on the safe

Correct Answer: A

The most likely cause for this issue is A. The application ID or Application Provider does not have the correct permissions on the safe. The CyberArk Central Credential Provider (CCP) is a web service that enables applications to retrieve secrets from the CyberArk Vault using REST API calls. The CCP requires an application ID or an Application Provider to authenticate and authorize the application before returning the requested secret. The application ID or Application Provider must have the Retrieve and List permissions on the safe where the secret is stored, otherwise the CCP will not be able to find the matching secret and will return an error. To resolve this issue, you should verify that the application ID or Application Provider has the correct permissions on the safe, and that the safe name and object name are correctly specified in the REST API call. You can use the CyberArk Privileged Access Security Web Access (PVWA) or the PrivateArk Client to check and modify the permissions on the safe. You can also use the CyberArk REST API Tester or a tool like Postman to test the REST API call and see the response from the CCP. For more information, refer to the following resources: Credential Providers - Centralized Credential Management | CyberArk, Section "Central Credential Provider" Credential Provider - CyberArk, Section "Using the Credential Provider" How to Build Your Secrets Management REST API\\'s into Postman, Section "How to Build Your Secrets Management REST API\\'s into Postman"

QUESTION 5

Which statement is true for the Conjur Command Line Interface (CLI)?

A. It is supported on Windows, Red Hat Enterprise Linux, and macOS.

B. It can only be run from the Conjur Leader node.

C. It is required for working with the Conjur REST API.

D. It does not implement the Conjur REST API for managing Conjur resources.

Correct Answer: A

This is the correct answer because the Conjur CLI is a tool that allows users to interact with the Conjur REST API from the command line. The Conjur CLI can be run on Windows, Red Hat Enterprise Linux, and macOS operating systems, as well as in Docker containers. The Conjur CLI can be installed using various methods, such as downloading the executable file, using a package manager, or pulling the Docker image. The Conjur CLI supports Conjur Enterprise 12.9 or later versions. This answer is based on the CyberArk Secrets Manager documentation1 and the CyberArk Secrets Manager training course2. The other options are not true statements for the Conjur CLI. The Conjur CLI can be run from any machine that has network access to the Conjur server, not only from the Conjur Leader node. The Conjur Leader node is the node that performs read/write operations on the Conjur database and policy engine, and hosts the Conjur UI and API endpoints. The Conjur CLI is not required for working with the Conjur REST API, as users can also use other tools, such as curl, Postman, or web browsers, to send HTTP requests to the Conjur REST API. The Conjur CLI does implement the Conjur REST API for managing Conjur resources, such as roles, policies, secrets, and audit records. The Conjur CLI provides a set of commands that correspond to the Conjur REST API endpoints and allow users to perform various operations on the Conjur resources.