VAULT-ASSOCIATE$^{Q\&As}$

HashiCorp Certified: Vault Associate (002)

Pass HashiCorp VAULT-ASSOCIATE Exam with 100% Guarantee

QUESTION 1

Which of the following statements describe the CLI command below?

S vault login -method-1dap username-mitche11h

A. Generates a token which is response wrapped

B. You will be prompted to enter the password

C. By default the generated token is valid for 24 hours

D. Fails because the password is not provided

Correct Answer: A

The CLI command vault login -method ldap username=mitchellh generates a token that is response wrapped. This means that the token contains a base64-encoded response wrapper, which is a JSON object that contains information about the token, such as its policies, metadata, and expiration time. The response wrapper is used to verify the authenticity and integrity of the token, and to prevent replay attacks. The response wrapper also allows Vault to automatically renew the token when it expires, or to revoke it if it is compromised. The -method ldap option specifies that the authentication method is LDAP, which requires a username andpassword to be provided. The username mitchellh is an example of an LDAP user name, and the password will be hidden when entered. References: Vault CLI Reference | Vault | HashiCorp Developer, Vault CLI Reference | Vault | HashiCorp Developer

QUESTION 2

What is the Vault CLI command to query information about the token the client is currently using?

A. vault lookup token

B. vault token lookup

C. vault lookup self

D. vault self-lookup

Correct Answer: B

The Vault CLI command to query information about the token the client is currently using is vault token lookup. This command displays information about the token or accessor provided as an argument, or the locally authenticated token if no argument is given. The information includes the token ID, accessor, policies, TTL, creation time, and metadata. This command can be useful for debugging and auditing purposes, as well as for renewing or revoking tokens. References: token lookup - Command | Vault | HashiCorp Developer, Tokens | Vault | HashiCorp Developer

QUESTION 3

Which of the following cannot define the maximum time-to-live (TTL) for a token?

A. By the authentication method t natively provide a method of expiring credentials

B. By the client system f credentials leaking

C. By the mount endpoint configurationvery password used

D. A parent token TTL e password rotation tools and practices

E. System max TTL

Correct Answer: B

The maximum time-to-live (TTL) for a token is defined by the lowest value among the following factors:

The authentication method that issued the token. Each auth method can have a default and a maximum TTL for the tokens it generates. These values can be configured by the auth method\\'s mount options or by the auth method\\'s specific

endpoints.

The mount endpoint configuration that the token is accessing. Each secrets engine can have a default and a maximum TTL for the leases it grants. These values can be configured by the secrets engine\\'s mount options or by the secrets

engine\\'s specific endpoints.

A parent token TTL. If a token is created by another token, it inherits the remaining TTL of its parent token, unless the parent token has an infinite TTL (such as the root token). A child token cannot outlive its parent token. System max TTL.

This is a global limit for all tokens and leases in Vault. It can be configured by the system backend\\'s max_lease_ttl option. The client system that uses the token cannot define the maximum TTL for the token, as this is determined by Vault\\'s

configuration and policies. The client system can only request a specific TTL for the token, but this request is subject to the limits imposed by the factors above.

References: https://developer.hashicorp.com/vault/docs/concepts/tokens3, https://developer.hashicorp.com/vault/docs/concepts/lease2, https://developer.hashicorp.com/vault/docs/commands/auth/tune4, https://developer.hashicorp.com/

vault/docs/commands/secrets/tune5, https://developer.hashicorp.com/vault/docs/commands/token/create6

QUESTION 4

An organization would like to use a scheduler to track and revoke access granted to a job (by Vault) at completion. What auth-associated Vault object should be tracked to enable this behavior?

A. Token accessor

B. Token ID

C. Lease ID

D. Authentication method

Correct Answer: C

A lease ID is a unique identifier that is assigned by Vault to every dynamic secret and service type authentication token. A lease ID contains information such as the secret path, the secret version, the secret type, etc. A lease ID can be used

to track and revoke access granted to a job by Vault at completion, as it allows the scheduler to perform the following operations:

Lookup the lease information by using the vault lease lookup command or the sys/leases/lookup API endpoint. This will return the metadata of the lease, such as the expire time, the issue time, the renewable status, and the TTL. Renew the

lease if needed by using the vault lease renew command or the sys/leases/renew API endpoint. This will extend the validity of the secret or the token for a specified increment, or reset the TTL to the original value if no increment is given.

Revoke the lease when the job is completed by using the vault lease revoke command or the sys/leases/revoke API endpoint. This will invalidate the secret or the token immediately and prevent any further renewals. For example, with the

AWS secrets engine, the access keys will be deleted from AWS the moment a lease is revoked.

A lease ID is different from a token ID or a token accessor. A token ID is the actual value of the token that is used to authenticate to Vault and perform requests. A token ID should be treated as a secret and protected from unauthorized

access. A token accessor is a secondary identifier of the token that is used for token management without revealing the token ID. A token accessor can be used to lookup, renew, or revoke a token, but not to authenticate to Vault or access

secrets. A token ID or a token accessor can be used to revoke the token itself, but not the leases associated with the token. To revoke the leases, a lease ID is required.

An authentication method is a way to verify the identity of a user or a machine and issue a token with appropriate policies and metadata. An authentication method is not an object that can be tracked or revoked, but a configuration that can be

enabled, disabled, tuned, or customized by using the vault auth commands or the sys/auth API endpoints. References: (https://developer.hashicorp.com/vault/docs/commands/lease/lookup), (https://developer.hashicorp.com/vault/docs/

commands/lease/renew), (https://developer.hashicorp.com/vault/docs/commands/lease/revoke), (https://developer.hashicorp.com/vault/docs/concepts/tokens#token-accessors), (https://developer.hashicorp.com/vault/docs/concepts/auth)

QUESTION 5

What is a benefit of response wrapping?

A. Log every use of a secret

B. Load balanc secret generation across a Vault cluster

C. Provide error recovery to a secret so it is not corrupted in transit

D. Ensure that only a single party can ever unwrap the token and see what\\'s inside

Correct Answer: D

Response wrapping is a feature that allows Vault to take the response it would have sent to a client and instead insert it into the cubbyhole of a single-use token, returning that token instead. The client can then unwrap the token and retrieve

the original response. Response wrapping has several benefits, such as providing cover, malfeasance detection, and lifetime limitation for the secret data. One of the benefits is to ensure that only a single party can ever unwrap the token and see what\\'s inside, as the token can be used only once and cannot be unwrapped by anyone else, even the root user or the creator of the token. This provides a way to securely distribute secrets to the intended recipients and detect any tampering or interception along the way5. The other options are not benefits of response wrapping: Log every use of a secret: Response wrapping does not log every use of a secret, as the secret is not directly exposed to the client or the network. However, Vault does log the creation and deletion of the response-wrapping token, and the client can use the audit device to log the unwrapping operation6. Load balance secret generation across a Vault cluster: Response wrapping does not load balance secret generation across a Vault cluster, as the secret is generated by the Vault server that receives the request and the response- wrapping token is bound to that server. However, Vault does support high availability and replication modes that can distribute the load and improve the performance of the cluster7. Provide error recovery to a secret so it is not corrupted in transit: Response wrapping does not provide error recovery to a secret so it is not corrupted in transit, as the secret is encrypted and stored in the cubbyhole of the token and cannot be modified or corrupted by anyone. However, if the token is lost or expired, the secret cannot be recovered either, so the client should have a backup or retry mechanism to handle such cases. References:
5(https://developer.hashicorp.com/vault/docs/concepts/response- wrapping),
6(https://developer.hashicorp.com/vault/docs/secrets), 7(https://developer.hashi corp.com/vault/docs/secrets),
(https://developer.hashicorp.com/vault/ tutorials/secrets- management/cubbyhole-response-wrapping)