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

HashiCorp Certified: Vault Associate (002)

Pass HashiCorp VAULT-ASSOCIATE Exam with 100% Guarantee

QUESTION 1

When unsealing Vault, each Shamir unseal key should be entered:

A. Sequentially from one system that all of the administrators are in front of

B. By different administrators each connecting from different computers

C. While encrypted with each administrators PGP key

D. At the command line in one single command

Correct Answer: B

When unsealing Vault, each Shamir unseal key should be entered by different administrators each connecting from different computers. This is because the Shamir unseal keys are split into shares that are distributed to trusted operators, and no single operator should have access to more than one share. This way, the unseal process requires the cooperation of a quorum of key holders, and enhances the security and availability of Vault. The unseal keys can be entered via multiple mechanisms from multiple client machines, and the process is stateful. The order of the keys does not matter, as long as the threshold number of keys is reached. The unseal keys should not be entered at the command line in one single command, as this would expose them to the history and compromise the security. The unseal keys should not be encrypted with each administrator\\'s PGP key, as this would prevent Vault from decrypting them and reconstructing the master key. References: https://developer.hashicorp.com/vault/docs/concepts/seal3, https://developer.hashicorp.com/vault/docs/commands/operator/unseal

QUESTION 2

An authentication method should be selected for a use case based on:

A. The auth method that best establishes the identity of the client

B. The cloud provider for which the client is located on

C. The strongest available cryptographic hash for the use case

D. Compatibility with the secret engine which is to be used

Correct Answer: A

An authentication method should be selected for a use case based on the auth method that best establishes the identity of the client. The identity of the client is the basis for assigning a set of policies and permissions to the client in Vault. Different auth methods have different ways of verifying the identity of the client, such as using passwords, tokens, certificates, cloud credentials, etc. Depending on the use case, some auth methods may be more suitable or convenient than others. For example, for human users, the userpass or ldap auth methods may be easy to use, while for machines or applications, the approle or aws auth methods may be more secure and scalable. The choice of the auth method should also consider the trade-offs between security, performance, and usability. References: Auth Methods | Vault | HashiCorp Developer, Authentication - Concepts | Vault | HashiCorp Developer

QUESTION 3

You are performing a high number of authentications in a short amount of time. You\\'re experiencing slow throughput for token generation. How would you solve this problem?

A. Increase the time-to-live on service tokens

B. Implement batch tokens

C. Establish a rate limit quota

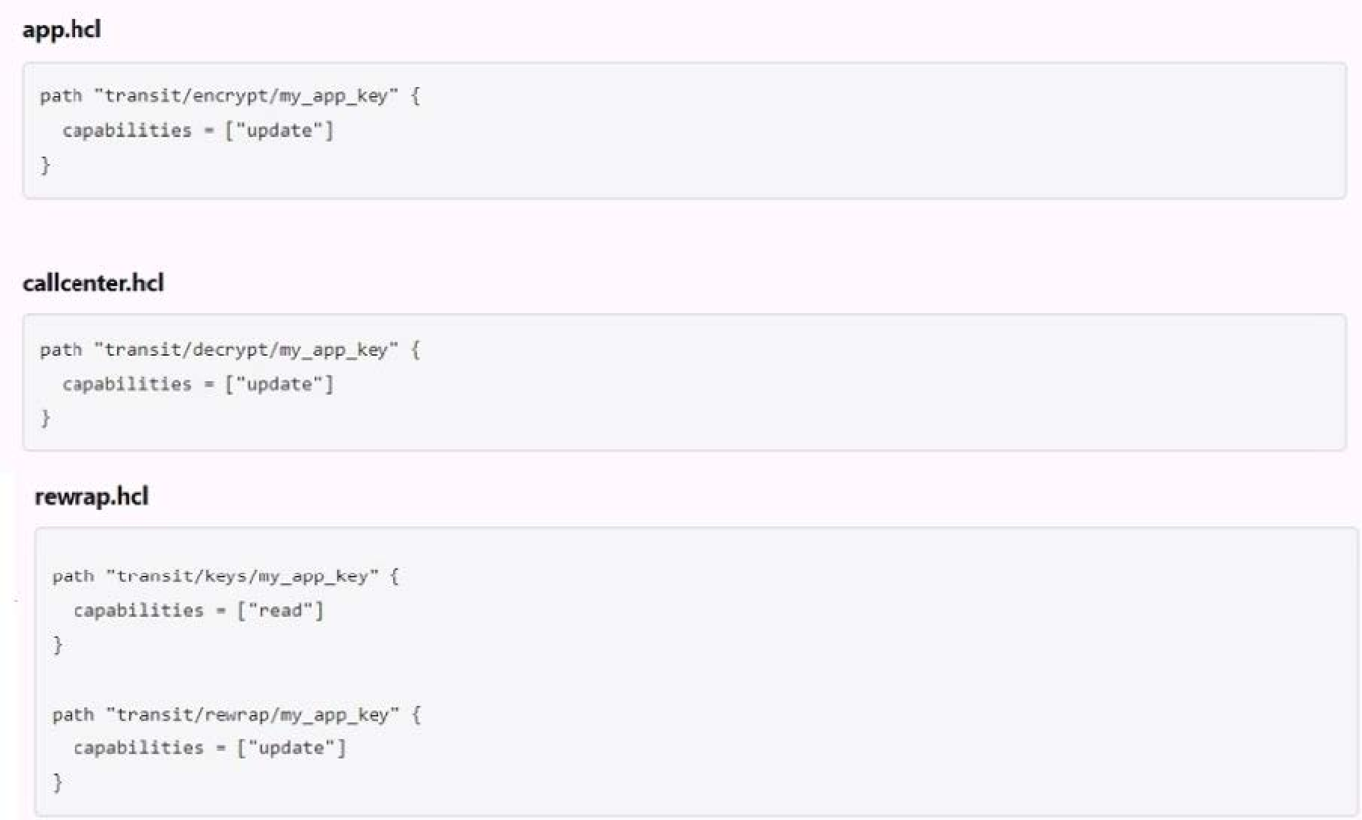
D. Reduce the number of policies attached to the tokens

Correct Answer: B

Batch tokens are a type of tokens that are not persisted in Vault\\'s storage backend, but are encrypted blobs that carry enough information to perform Vault actions. Batch tokens are extremely lightweight and scalable, and can improve the throughput for token generation. Batch tokens are suitable for high-volume and ephemeral workloads, such as containers or serverless functions, that require short-lived and non-renewable tokens. Batch tokens can be created by using the type=batch flag in the vault token create command, or by configuring the token_type parameter in the auth method\\'s role or mount options. Batch tokens have some limitations compared to service tokens, such as the lack of renewal, revocation, listing, accessor, and cubbyhole features. Therefore, batch tokens should be used with caution and only when the trade-offs are acceptable. References: https://developer.hashicorp.com/vault/tutorials/tokens/batch-tokens1, https://developer.hashicorp.com/vault/docs/commands/token/create2, https://developer.hashicorp.com/vault/docs/concepts/tokens#token-types3

QUESTION 4

The following three policies exist in Vault. What do these policies allow an organization to do?



app.hcl

```
path "transit/encrypt/my_app_key" {
  capabilities = ["update"]
}
```

callcenter.hcl

```
path "transit/decrypt/my_app_key" {
  capabilities = ["update"]
}
```

rewrap.hcl

```
path "transit/keys/my_app_key" {
  capabilities = ["read"]
}

path "transit/rewrap/my_app_key" {
  capabilities = ["update"]
}
```

A. Separates permissions allowed on actions associated with the transit secret engine

B. Nothing, as the minimum permissions to perform useful tasks are not present

C. Encrypt, decrypt, and rewrap data using the transit engine all in one policy

D. Create a transit encryption key for encrypting, decrypting, and rewrapping encrypted data

Correct Answer: C

The three policies that exist in Vault are: admins: This policy grants full access to all secrets and operations in Vault. It can be used by administrators or operators who need to manage all aspects of Vault. default: This policy grants access to all secrets and operations in Vault except for those that require specific policies. It can be used as a fallback policy when no other policy matches. transit: This policy grants access only to the transit secrets engine, which handles cryptographic functions on data in-transit. It can be used by applications or services that need to encrypt or decrypt data using Vault. These policies allow an organization to perform useful tasks such as: Encrypting, decrypting, and rewrapping data using the transit engine all in one policy: This policy grants access to both the transit secrets engine and the default policy, which allows performing any operation on any secret in Vault. Creating a transit encryption key for encrypting, decrypting, and rewrapping encrypted data: This policy grants access only to the transit secrets engine and its associated keys, which are used for encrypting and decrypting data in transit using AES-GCM with a 256-bit AES key or other supported key types. Separating permissions allowed on actions associated with the transit secret engine: This policy grants access only to specific actions related to the transit secrets engine, such as creating keys or wrapping requests. It does not grant access to other operations or secrets in Vault.

QUESTION 5

When using Integrated Storage, which of the following should you do to recover from possible data loss?

A. Failover to a standby node

B. Use snapshot

C. Use audit logs

D. Use server logs

Correct Answer: B

Integrated Storage is a Raft-based storage backend that allows Vault to store its data internally without relying on an external storage system. It also enables Vault to run in high availability mode with automatic leader election and failover. However, Integrated Storage is not immune to data loss or corruption due to hardware failures, network partitions, or human errors. Therefore, it is recommended to use the snapshot feature to backup and restore the Vault data periodically or on demand. A snapshot is a point-in-time capture of the entire Vault data, including the encrypted secrets, the configuration, and the metadata. Snapshots can be taken and restored using the vault operator raft snapshot command or the sys/ storage/raft/snapshot API endpoint. Snapshots are encrypted and can only be restored with a quorum of unseal keys or recovery keys. Snapshots are also portable and can be used to migrate data between different Vault clusters or storage backends. References:
https://developer.hashicorp.com/vault/docs/concepts/integrated- storage1,
https://developer.hashicorp.com/vault/docs/commands/operator/raft/snapshot2,
https://developer.hashicorp.com/vault/api-docs/system/ storage/raft/snapshot3

QUESTION 6

A web application uses Vault\\'s transit secrets engine to encrypt data in-transit. If an attacker intercepts the data in transit which of the following statements are true? Choose two correct answers.

A. You can rotate the encryption key so that the attacker won\\'t be able to decrypt the data

B. The keys can be rotated and min_decryption_version moved forward to ensure this data cannot be decrypted

C. The Vault administrator would need to seal the Vault server immediately

D. Even if the attacker was able to access the raw data, they would only have encrypted bits (TLS in transit)

Correct Answer: BD

A web application that uses Vault\\'s transit secrets engine to encrypt data in- transit can benefit from the following security features: Even if the attacker was able to access the raw data, they would only have encrypted bits (TLS in transit). This means that the attacker would need to obtain the encryption key from Vault in order to decrypt the data, which is protected by Vault\\'s authentication and authorization mechanisms. The transit secrets engine does not store the data sent to it, so the attacker cannot access the data from Vault either. The keys can be rotated and min_decryption_version moved forward to ensure this data cannot be decrypted. This means that the web application can periodically change the encryption key used to encrypt the data, and set a minimum decryption version for the key, which prevents older versions of the key from being used to decrypt the data. This way, even if the attacker somehow obtained an old version of the key, they would not be able to decrypt the data that was encrypted with a newer version of the key. The other statements are not true, because: You cannot rotate the encryption key so that the attacker won\\'t be able to decrypt the data. Rotating the key alone does not prevent the attacker from decrypting the data, as they may still have access to the old version of the key that was used to encrypt the data. You need to also move the min_decryption_version forward to invalidate the old version of the key. The Vault administrator would not need to seal the Vault server immediately. Sealing the Vault server would make it inaccessible to both the attacker and the legitimate users, and would require unsealing it with the unseal keys or the recovery keys. Sealing the Vault server is a last resort option in case of a severe compromise or emergency, and is not necessary in this scenario, as the attacker does not have access to the encryption key or the data in Vault. References: Transit Secrets Engines | Vault | HashiCorp Developer, Encryption as a service: transit secrets engine | Vault | HashiCorp Developer

QUESTION 7

You can build a high availability Vault cluster with any storage backend.

A. True

B. False

Correct Answer: B

Not all storage backends support high availability mode for Vault. Only the storage backends that support locking can enable Vault to run in a multi-server mode where one server is active and the others are standby. Some examples of storage backends that support high availability mode are Consul, Integrated Storage, and ZooKeeper. Some examples of storage backends that do not support high availability mode are Filesystem, MySQL, and PostgreSQL. References: https://developer.hashicorp.com/vault/docs/concepts/ha1, https://developer.hashicorp.com/vault/docs/configuration/storage2

QUESTION 8

How would you describe the value of using the Vault transit secrets engine?

A. Vault has an API that can be programmatically consumed by applications

B. The transit secrets engine ensures encryption in-transit and at-rest is enforced enterprise wide

C. Encryption for application data is best handled by a storage system or database engine, while storing encryption keys in Vault

D. The transit secrets engine relieves the burden of proper encryption/decryption from application developers and pushes the burden onto the operators of Vault

Correct Answer: D

The transit secrets engine relieves the burden of proper encryption/decryption from application developers and pushes the burden onto the operators of Vault. The transit secrets engine provides encryption as a service, which means that it performs cryptographic operations on data in-transit without storing any data. This allows developers to delegate the responsibility of managing encryption keys and algorithms to Vault operators, who can define and enforce policies on the transit secrets engine. This way, developers can focus on their application logic and data, while Vault handles the encryption and decryption of data in a secure and scalable manner. References: Transit - Secrets Engines | Vault | HashiCorp Developer, Encryption as a service: transit secrets engine | Vault | HashiCorp Developer

QUESTION 9

Which Vault secret engine may be used to build your own internal certificate authority?

A. Transit

B. PKI

C. PostgreSQL D. Generic

Correct Answer: B

The Vault secret engine that can be used to build your own internal certificate authority is the PKI secret engine. The PKI secret engine generates dynamic X.509 certificates on-demand, without requiring manual processes of generating private keys and CSRs, submitting to a CA, and waiting for verification and signing. The PKI secret engine can act as a root CA or an intermediate CA, and can issue certificates for various purposes, such as TLS, code signing, email encryption, etc. The PKI secret engine can also manage the certificate lifecycle, such as rotation, revocation, renewal, and CRL generation. The PKI secret engine can also integrate with external CAs, such as Venafi or Entrust, to delegate the certificate issuance and management. References: PKI - Secrets Engines | Vault | HashiCorp Developer, Build Your Own Certificate Authority (CA) | Vault - HashiCorp Learn

QUESTION 10

Where does the Vault Agent store its cache?

A. In a file encrypted using the Vault transit secret engine

B. In the Vault key/value store

C. In an unencrypted file

D. In memory

Correct Answer: D

The Vault Agent stores its cache in memory, which means that it does not persist the cached tokens and secrets to disk or any other storage backend. This makes the cache more secure and performant, as it avoids exposing the sensitive data to potential attackers or unauthorized access. However, this also means that the cache is volatile and will be lost if the agent process is terminated or restarted. To mitigate this, the agent can optionally use a persistent cache file to restore the tokens and leases from a previous agent process. The persistent cache file is encrypted using a key derived from the agent\\'s auto-auth token and a nonce, and it is stored in a user-specified location on disk. References: Caching Vault Agent | Vault | HashiCorp Developer, Vault Agent Persistent Caching | Vault | HashiCorp Developer