VAULT-ASSOCIATE Q&As

HashiCorp Certified: Vault Associate (002)

Pass HashiCorp VAULT-ASSOCIATE Exam with 100% Guarantee

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

https://www.geekcert.com/vault-associate.html

100% Passing Guarantee
100% Money Back Assurance

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

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

QUESTION 1

Which of these are a benefit of using the Vault Agent?

A. Vault Agent allows for centralized configuration of application secrets engines

B. Vault Agent will auto-discover which authentication mechanism to use

C. Vault Agent will enforce minimum levels of encryption an application can use

D. Vault Agent will manage the lifecycle of cached tokens and leases automatically

Correct Answer: D

Vault Agent is a client daemon that provides the following features:

Auto-Auth - Automatically authenticate to Vault and manage the token renewal process for locally-retrieved dynamic secrets.

API Proxy - Allows Vault Agent to act as a proxy for Vault\\'s API, optionally using (or forcing the use of) the Auto-Auth token.

Caching - Allows client-side caching of responses containing newly created tokens and responses containing leased secrets generated off of these newly created tokens. The agent also manages the renewals of the cached tokens and

leases. Templating - Allows rendering of user-supplied templates by Vault Agent, using the token generated by the Auto-Auth step.

Process Supervisor Mode - Runs a child process with Vault secrets injected as environment variables.

One of the benefits of using the Vault Agent is that it will manage the lifecycle of cached tokens and leases automatically. This means that the agent will handle the token renewal and revocation logic, as well as the lease renewal and

revocation logic for the secrets that are cached by the agent. This reduces the burden on the application developers and operators, and ensures that the tokens and secrets are always valid and up-to-date. References: Vault Agent | Vault |

HashiCorp Developer, Caching - Vault Agent | Vault | HashiCorp Developer

QUESTION 2

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 3

What does the following policy do?

```
path "secret/data/{{identity.entity.id}}/*" {
  capabilities = ["create", "update", "read", "delete"]
}

path "secret/metadata/{{identity.entity.id}}/*" {
  capabilities = ["list"]
}
```

A. Grants access for each user to a KV folder which shares their id

B. Grants access to a special system entity folder

C. Allows a user to read data about the secret endpoint identity

D. Nothing, this is not a valid policy

Correct Answer: C

This policy allows a user to read data about the secret endpoint identity. The policy grants the user the ability to create, update, read, and delete data in the "secret/data/{identity.entity.id}" path. Additionally, the user is allowed to list data in the

"secret/metadata/{identity.entity.id}" path. This policy is useful for users who need to access information about the secret endpoint identity.

The secret endpoint identity is a feature of the Identity Secrets Engine, which allows Vault to generate identity tokens that can be used to access other Vault secrets engines or namespaces. The identity tokens are based on the entity and

group information of the user or machine that authenticates with Vault. The entity is a unique identifier for the user or machine, and the group is a collection of entities that share some common attributes. The identity tokens can carry

metadata and policies that are associated with the entity and group.

The "secret/data/{identity.entity.id}" path is where the user can store and retrieve data that is related to the secret endpoint identity. For example, the user can store some configuration or preferences for the secret endpoint identity in this path.

The "secret/metadata/{identity.entity.id}" path is where the user can list the metadata of the data stored in the "secret/data/{identity.entity.id}" path. For example, the user can list the version, creation time, deletion time, and destroy

time of the

data in this path.

References:

[Identity - Secrets Engines | Vault | HashiCorp Developer] [KV - Secrets Engines | Vault | HashiCorp Developer]

QUESTION 4

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

QUESTION 5

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