Vault – Test Automation

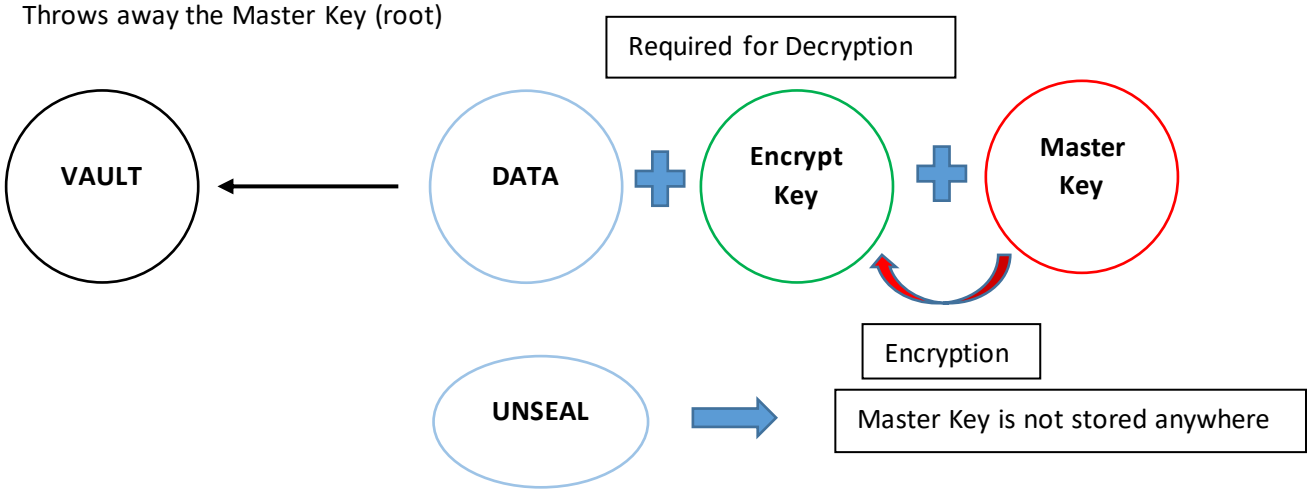
-----> Concept ->

Uses Cases:

- *General Secret Storage – (Env, Cfg) – Vault Read*
- *Employee Credentials Storage – Web Access*
- *API Key Gen – AWS Key*
- *Data Encryption*

Operations:

- Seal – No Operations Possible – Decryption not possible
- Un-Seal – Constructing MASTER Key – Decryption Key
- MASTER KEY: Not Stored Anywhere – Used to Obtain Encryption Key – Constructed when UNSEAL is executed – Shamir Secret into Shards – Until Server restart or API Seal Call – Seal Throws away the Master Key (root)



-----> TEST Environment ->

- ➔ Linux Server (Virutal Box)
- ➔ Vault Setup
- ➔ Client – Telnet, SSH, HTTP API
- ➔ Language – Python

-----> TEST Cases - - ->

- General Secret Storage (Basic Test Case)

STEPS:

1. ***Initialize the Vault***
   The vault should be initialized with the key share and threshold
   Vault init -key-shares=1 -key-threshold=1

Output :

```
$ vault init -key-shares=1 -key-threshold=1
Key 1:
bd40c6fe42a475fd4947f1f1996a74c420c843dc558454a23198f1883f122949
Initial Root Token: 0c7670fd-1fe7-ef17-5237-80d42ec37064

Vault initialized with 1 keys and a key threshold of 1. Please
securely distribute the above keys. When the Vault is re-sealed,
restarted, or stopped, you must provide at least 1 of these keys
to unseal it again.

Vault does not store the master key. Without at least 1 keys,
your Vault will remain permanently sealed.
```

2. ***UnSeal Vault***
   The vault should be unsealed to operate with it (read/write)
   Vault unseal <key>

Output:

```
$ vault unseal
bd40c6fe42a475fd4947f1f1996a74c420c843dc558454a23198f1883f122949
Sealed: false
Key Shares: 1
Key Threshold: 1
Unseal Progress: 0
```

3. ***Authorize your Request***
   The vault authorizes the identity of the client
   Vault auth <token>

Output:

```
vault auth 0c7670fd-1fe7-ef17-5237-80d42ec37064

Successfully authenticated!
token: 0c7670fd-1fe7-ef17-5237-80d42ec37064
token_duration: 0
token_policies: [root]
```

4. ***Write, Read and Printing Secret***

   Data can be write and read from the vault in the unsealed state

   Vault write <path> <value>

   Vault read <path>

Output:

```
$ vault write secret/hello vaule=world
Success! Data written to: secret/hello
```

```
$ vault read secret/hello
Key             Value
lease_duration  2592000
vaule           world
```