Working with the Transit Engine

Ned Bellavance
Founder, Ned in the Cloud LLC

@ned1313 | nedinthecloud.com

Overview

Introduce the Transit engine

Managing cryptographic keys

Applying data encryption

Transit Engine

Transit Engine

Encryption as a service

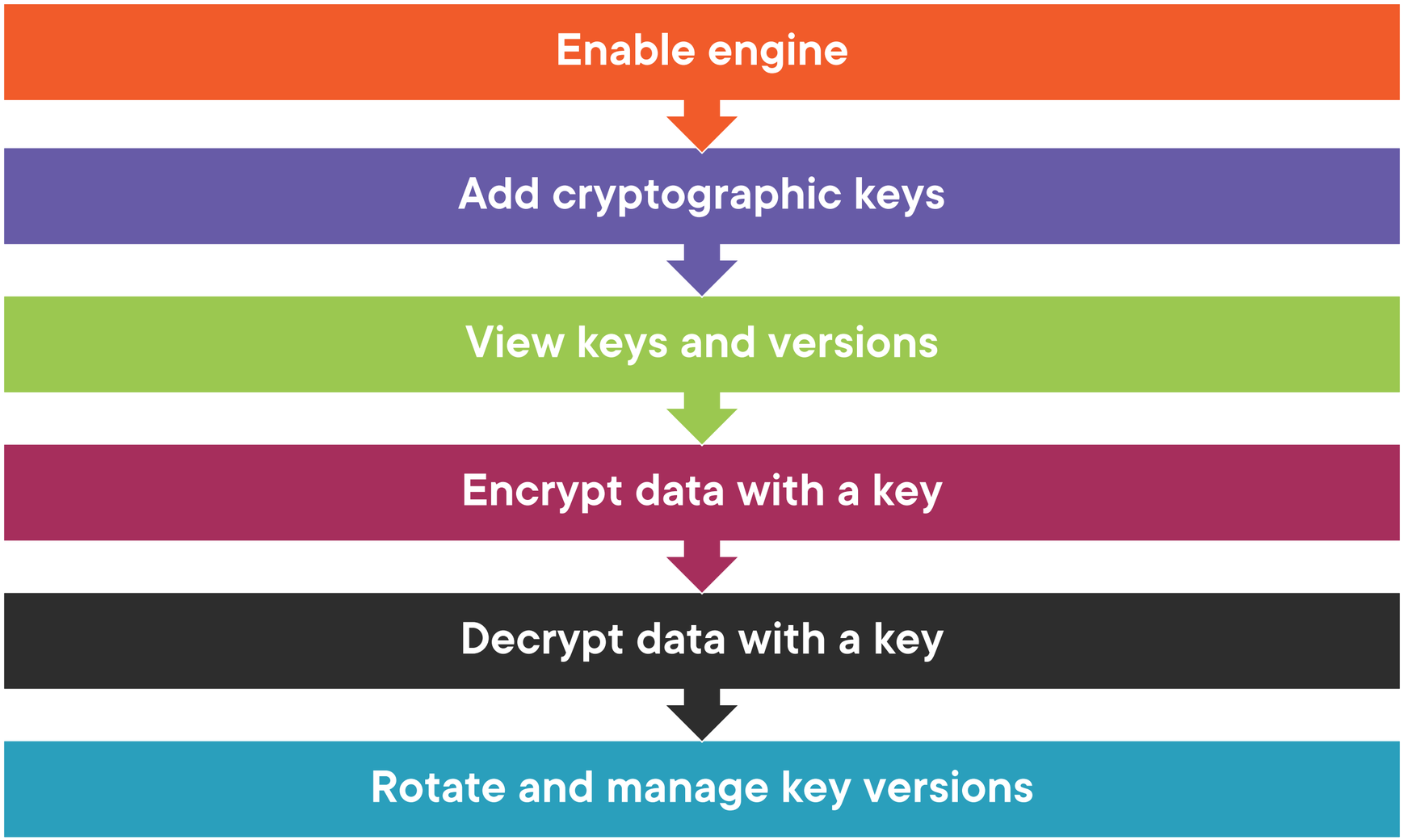
Does not store data

Engine manages keys

Supported actions:

- **Encrypt/decrypt**
- **Sign and verify**
- **Generate hashes**
- **Create random bytes**

Transit Operations



Globomantics Scenario

Use Case

- Application developer needs to encrypt data written to object storage
- Data will be generated by application
- Vault does not need to store data

Solution

- Enable an instance of the Transit engine
- Create keys for developers to use to encrypt data

Managing Keys

Key Features

Basic types

- AES, ChaCha20, ED25519, ECDSA, RSA

Type determines supported actions

Convergent encryption

- Deterministic
- Less secure

Key Versions

Each key is a map of keys

Working set are available

Retired keys stored in archive

Version settings

- min_decryption_version
- min_encryption_version

rotate **to add new versions**

config **and** trim **to manage versions**

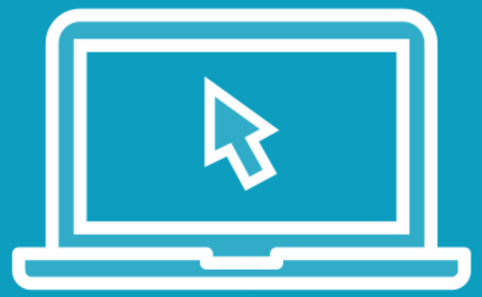
Working with the Transit Engine

```
# Create a key

vault write [-force] PATH/<keyname> [DATA] [parameters]

vault write -force transit/keys/ccid


# List keys

vault list PATH

vault list transit/keys


# Read key info

vault read PATH/<keyname>

vault read transit/keys/ccid
```

Working with the Transit Engine

```
# Rotate a key

vault write [-force] PATH/rotate [DATA]

vault write -force transit/keys/ccid/rotate


# Configure available versions

vault write PATH/config [parameters]

vault write transit/keys/ccid/config min_decryption_version=4


# Remove older versions

vault write PATH/trim [parameters]

vault write transit/keys/ccid/trim min_available_version=4
```

Demo



Tasks:

- **Enable Transit engine**
- **Create cryptographic key**
- **Encrypt and decrypt data**
- **Rotate keys and manage versions**

Encrypting Data

Encrypt and Decrypt Data

Value is base64 encoded

Ciphertext includes metadata

Uses current version

Value is base64 encoded

Version in ciphertext metadata

Key version may not be available

Encrypt and Decrypt Data

```
# Encrypt plaintext data

vault write PATH/encrypt/<keyname> [parameters]

vault write transit/encrypt/ccid plaintext=<base64 encoded value>


# Decrypt ciphertext data

vault write PATH/decrypt/<keyname> [parameters]

vault write transit/decrypt/ccid ciphertext=<ciphertext value>
```

Key Takeaways

The Transit engine provides encryption as a service with versioned cryptographic keys. Data is never stored in Vault.

Key types support different cryptographic actions, including encryption, signing, and hashing.

Cryptographic keys are composed of a working set and archive set. The working set is stored in memory.

Data submitted for encryption must be base64 encoded. The current version key will be used by default.

The key for decryption operations must be in the working set.

Up Next: Using the Vault Agent