Vault

Implementation Foundations

Module 9 Tokens

What You Will Learn

Authentication

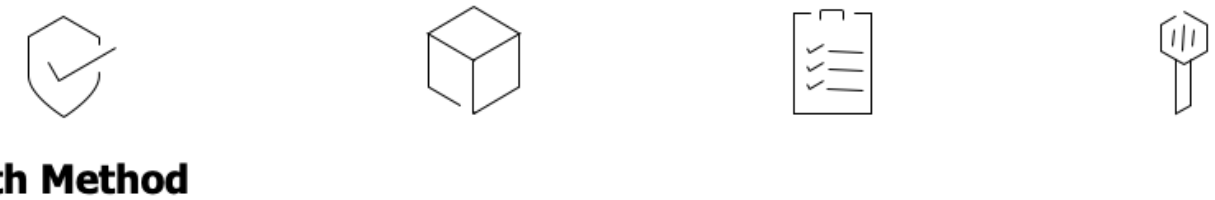
- Authentication Workflow
- Types of Authentication
- Introduction to Policies

Tokens

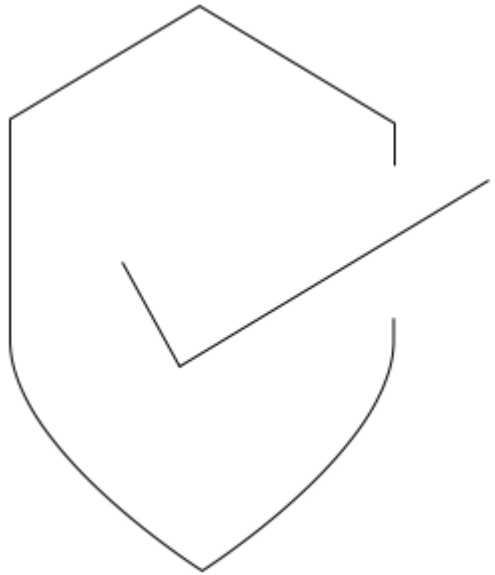
- Token Overview
- Token Types
- Token Lifecycle
- Token Use Cases

Authentication

Token Workflow - Authenticate



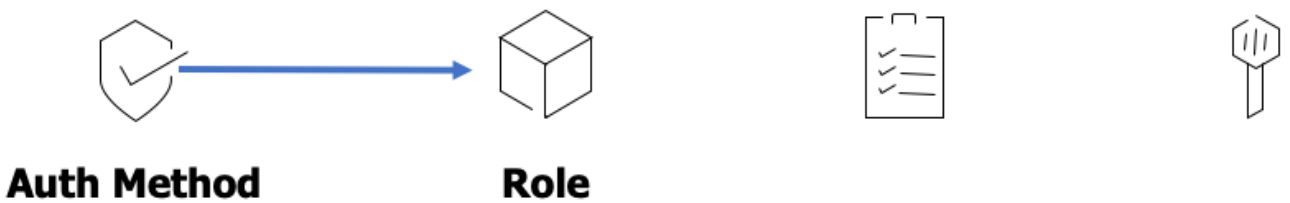
Vault Authentication Method



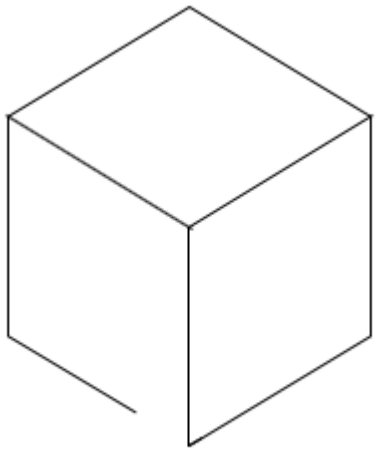
Authentication:

- The first step in gaining a token
- Leveraging trusted 3rd party source
- Validating identity only

Token Workflow - Role



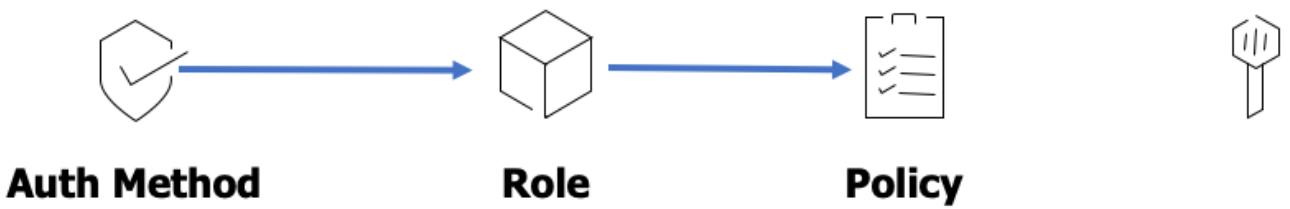
Vault Role



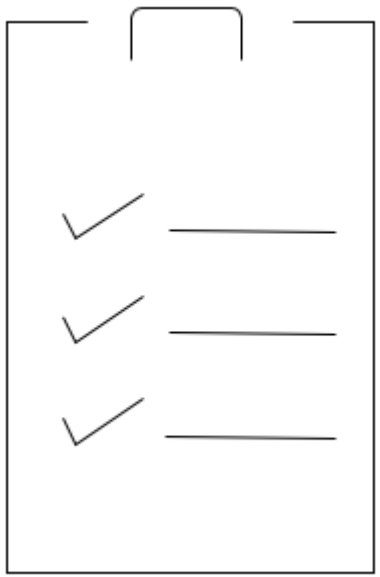
Vault Role:

- The link between an authentication method and policy
- Defines the endpoint queried for a token
- Mostly defined for dynamic secrets

Token Workflow - Policy



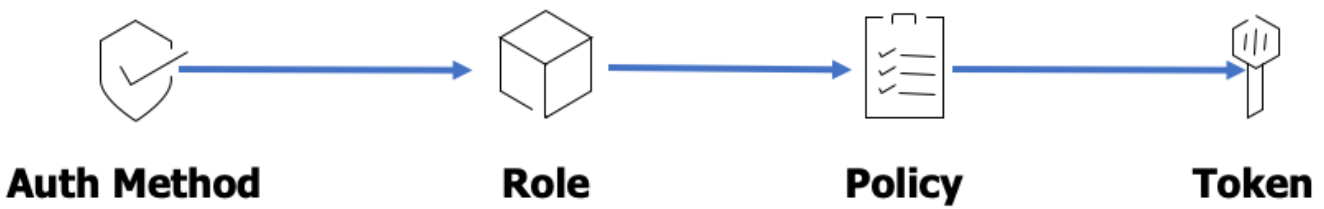
Vault Policy



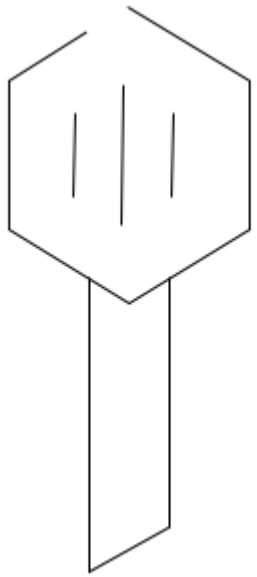
Vault Policy:

- Through the role a policy is linked
- Can have one or multiple policies
- Policies are additive in nature

Token Workflow - Token



Vault Token



Vault Token:

- Built-in and automatically available at /auth/token
- When any other auth methods returns an identity a token is generated
- A token is unique for that identity
- Bound to specifics of the role (TTL, No. of uses)
- The basis of renewal and revocation for all other methods

Tokens

Token Properties - Accessor

Token Accessor:

- Look up a token's properties
- Look up a token's capabilities on path
- Renew the token
- Revoke the token

```
Key                 Value
---                 -----
token               password
token_accessor      yUZY7mSGFsL8hcHNfw6Rtev2
token_duration      ∞
token_renewable     false
token_policies      ["root"]
identity_policies   []
policies            ["root"]
```

Token Properties - Time To Live

Token Duration:

- The full time duration of a token's life**
- Duration predicated based on renew-ability
- System default max is 32 days

```
Key                 Value
---                 -----
token               password
token_accessor      yUZY7mSGFsL8hcHNfw6Rtev2
token_duration      ∞
token_renewable     false
token_policies      ["root"]
identity_policies   []
policies            ["root"]
```

**This shows a root token
If the parent token is revoked the child wil be revoked regardless of the TTL

Token Properties - Policies

Token Policies:

- Policies are the permissions an individual token has in Vault
- Policies grant or forbid access to certain paths and operations in Vault
- Written in HCL
- Can have multiple policies on a token

```
Key                  Value
---                  -----
token                password
token_accessor       yUZY7mSGFsL8hcHNfw6Rtev2
token_duration       ∞
token_renewable      false
token_policies       ["root"]
identity_policies    []
policies             ["root"]
```

This shows a root policy

Token Auth Backend

```
$ vault token create -policy=umbrella-policy -policy=raccoon-policy
Key                  Value
--                    --
token                95eba8ed-f6fc-958a-f490-c7fd0eda5e9e
token_accessor       882d4a40-3796-d06e-c4f0-604e8503750b
token_duration       768h
token_renewable      true
token_policies       [umbrella-policy raccoon-policy]
```

- Invoked when another Auth method is used
- Can be used to explicitly create tokens
- Good for testing policies, checking permissions, development life-cycle
- Authentication bypass

Token Auth Backend

```
$ vault token create -policy=umbrella-policy -policy=raccoon-policy
Key                  Value
--                    --
token                95eba8ed-f6fc-958a-f490-c7fd0eda5e9e
token_accessor       882d4a40-3796-d06e-c4f0-604e8503750b
token_duration       768h
token_renewable      true
token_policies       [umbrella-policy raccoon-policy]
```

- Creating a token with two policies associated with it
- umbrella-policy and raccoon-policy are associated with this token

Lifecycle of a Token

Parent and Child Tokens:

- When a new token or secret is created, it is a child of the creator.
- If the parent is revoked or expires, so do all its children regardless of their own leases
- A child may be a token, secret, or authentication created by a parent
- An orphan token can be created to create token not bound by the parent

```
b519c6aa... (3h)
    6a2cf3e7... (4h)
    1d3fd4b2... (1h)
        794b6f2f... (2h)
```

Token Types

Service Token

- Service tokens are what are created by default
- They do not exist until request
- These are generally used for long use
- Not replicated over a performance link
- Adhere to the normal properties (TTL, Child/Parent)

Batch Token

- Bare minimum properties to be used to access secrets
- Fixed TTL
- Short use tokens
- Replicated across performance link
- Used in cases where systems need high frequency, short lived access to secrets

Token Differences

Functionality	Service Tokens	Batch Tokens
Can Be Root Tokens	Yes	No
Can Create Child Tokens	Yes	No
Can be Renewable	Yes	No
Can be Periodic	Yes	No
Can have Explicit Max TTL	Yes	No (always uses a fixed TTL)
Has Accessors	Yes	No
Has Cubbyhole	Yes	No
Revoked with Parent (if not orphan)	Yes	Stops Working
Dynamic Secrets Lease Assignment	Self	Parent (if not orphan)
Can be Used Across Performance Replication Clusters	No	Yes (if orphan)
Creation Scales with Performance Standby Node Count	No	Yes
Cost	Heavyweight	Lightweight

When to use a Token

- Embedded in applications
  - Used to retrieve secrets at runtime or provide auth material
- Used as authentication for automation
  - Used by a pipeline to generate new secrets for infrastructure or applications
- Part of a development lifecycle
  - Testing application development, implementing automation, securing organizations

When to use a Token (cont.)

- Tokens can be given a policy
  - What secrets can I access?
- Tokens can be given a TTL
  - How long do I live before needing rotation?
- Tokens can be obfuscated by passing only the accessor
  - You don't have the real token but you can look up properties of it (great for orchestrators)
- Tokens can be bound by cidr_ip
  - Restricts usage of the generated token to client IPs falling within the range of the specified CIDRs
- Tokens can be wrapped

Chapter Summary

- Authentication is a combination of Token Store and an Authentication Method
- Vault supports many "native" authentication methods
- Tokens have many important properties to be aware of
- There are built-in safety features on tokens

Reference links

- Token Authentication Method
- Getting Started with Tokens
- Concept Review of Tokens

Vault Tokens Module Complete!