Vault

Implementation Foundations

Module 15
Onboarding Applications & Users

What You Will Learn

- Operational Readiness
- Namespaces
- User/Service Onboarding
- Vault Service Usage Patterns

Operational Readiness

Operational Readiness Overview

Requirements Gathering

What are the business needs?

Vault Architecture Design

How will this work in the environment?

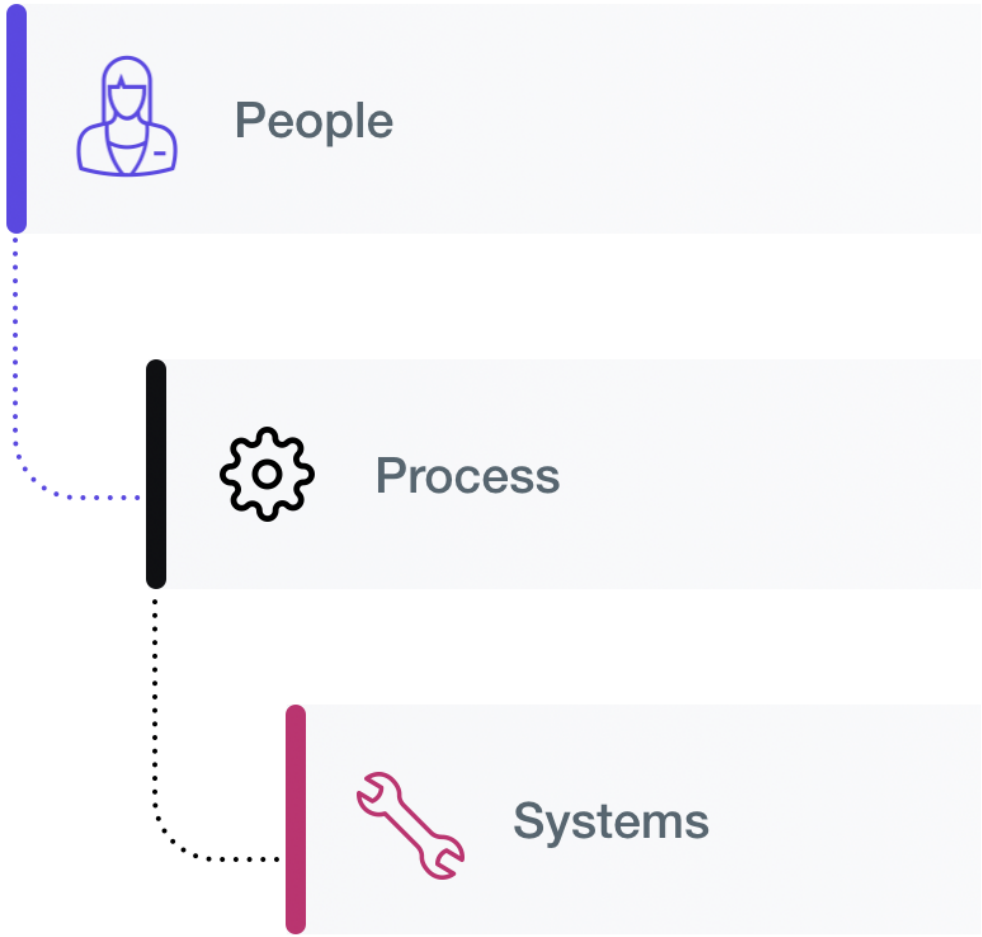
Vault Support Team Design

Who is responsible for availibility?

Vault Deployment Planning

What is the launch plan and delivery timeline?

Understanding The Business Need



People
What skills are needed?

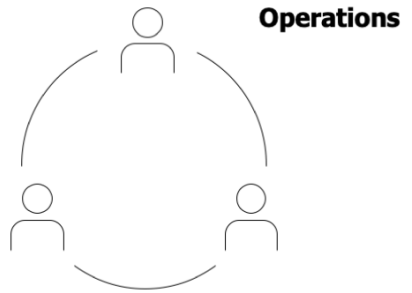
Process
How is the workflow going to change?

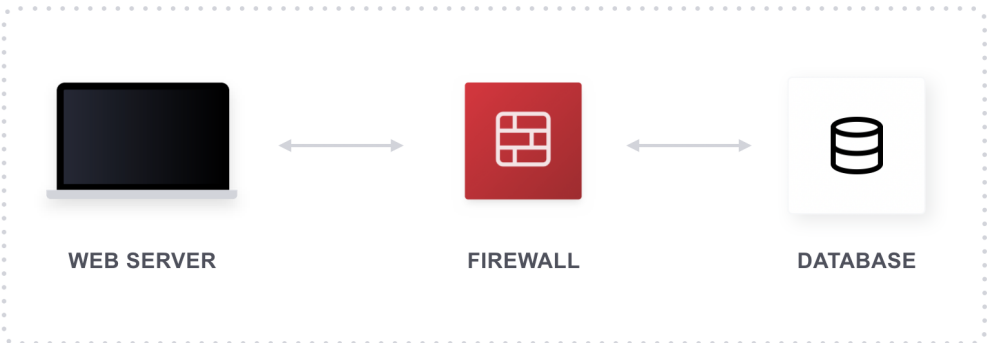
Technology
How to link people and process for an outcome.

Operational Concern

Operations responsibility :

Availability
Reliability
Access Controls

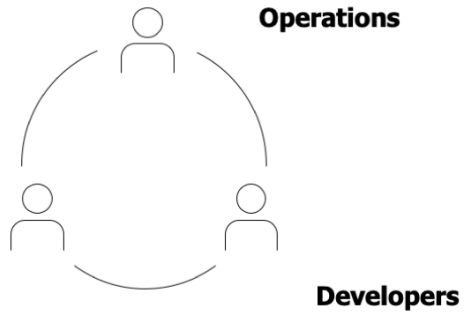


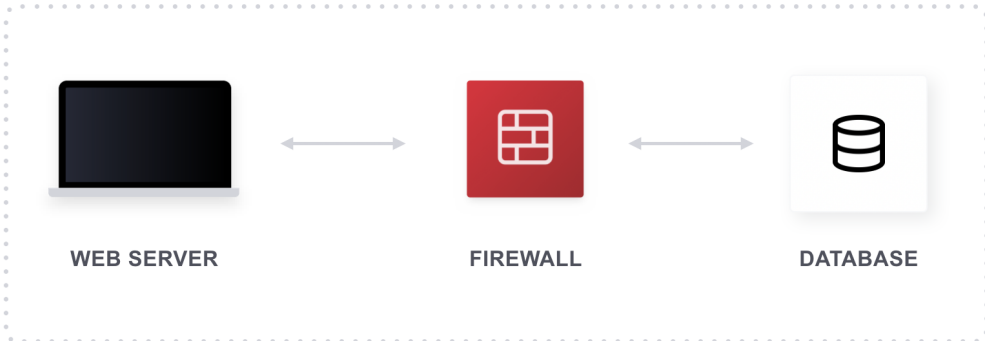


Development Concern

Developer responsibility :

Ease of access
Functional authorization
Low maintenance

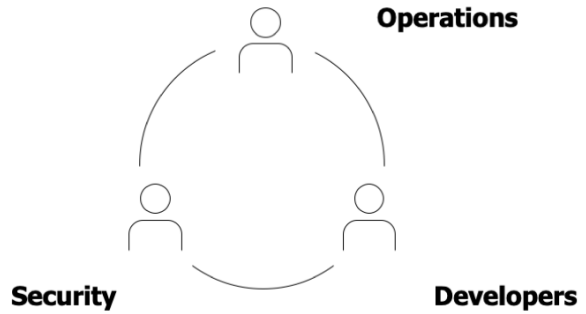


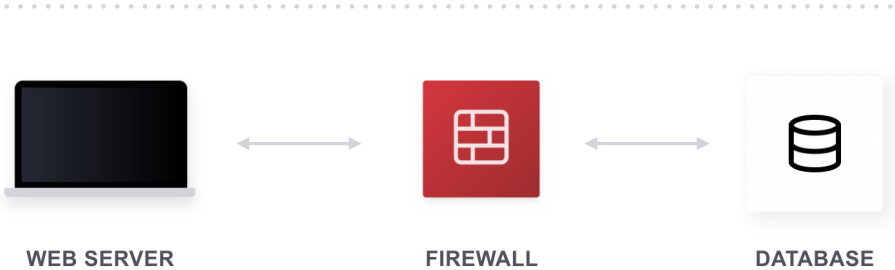


Security Concern

Security responsibility :

High audit output
Strong passwords
Granular access controls





Who Owns Vault?

Operations?

Who Owns Vault?

Operations?

Security?

Policy Owners and Service Owners

Operations Service Owners

- Reliability of Vault
- Authentication Methods
- Operational Access
- DR

Operational Concerns For Vault

Considerations

- What environments will be serviced?
- What clouds will be used?
- Network boundaries or configurations that may hinder access?
- What are the infrastructure resources needed?
- Onboarding workflows (People and Application based)

Security Service Owners

Security Service Owners

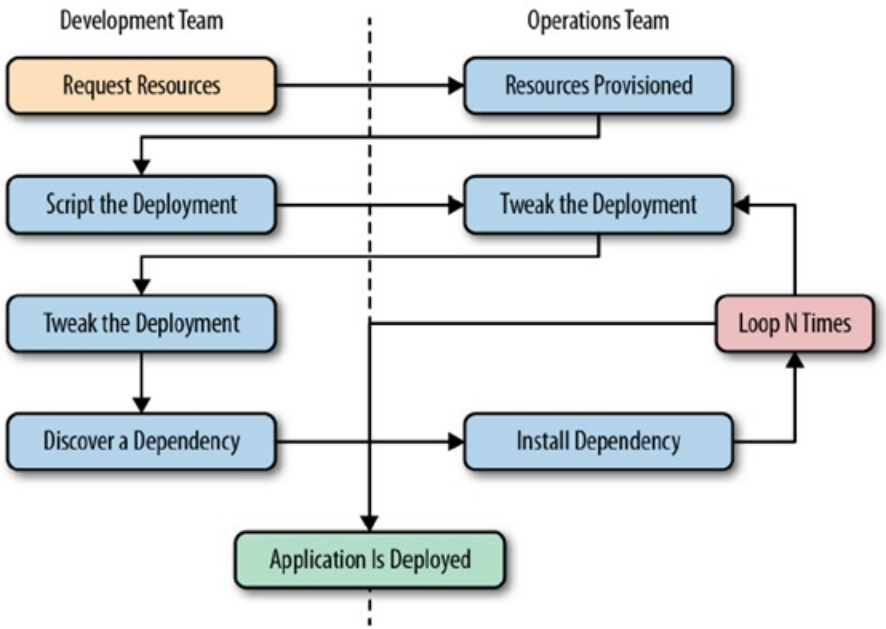
- Authorization Methods
- Secrets Policy
- Rotation Strategy
- Data, Application and Systems Access

Secure Access Concerns

Considerations

- What are the secrets lifecycle requirements?
- What are the prescribed authentication methods?
- What are the access patterns?
- What are the audit retention requirements?

Vault Service Consumers



Security Consumers

- Application Developer Enablement
- Infrastructure Access Workflows
- Cloud Access Controls

What Skills are Needed?

A Vault support team should have three role profiles:

Site Reliability Engineer

Sr. DevOps Engineer

DevOps Engineer

Site Reliability Engineer

Skill Set

- CI Tooling and Workflows
- Cloud and On-Premise Operations
- Version Control Systems
- Managing Shared Services

DevOps Engineer

Skill Set

- Configuration and Automations Tools
- Cloud and On-Prem Operations
- Deep Understanding of Software Development Lifecycles
- Managing Shared Services

Service Onboarding

Service Onboarding Checklist

- User and service requirements should be well documented
- Current secrets delivery model understood for service/user base
- Defined new secrets consumption model
- Define enablement process and communications

Determining User Needs

- Understand how users are consuming secrets
- Understand how do applications leverage access
- Understand the system management for vault agent deployment

Discovery Questions

- What is the spectrum of technical ability within my user base?
- Will they need a GUI or are they able to use the CLI?
- What authentication methods are required for the user base to authenticate with Vault?
- Are there accommodations that need to be made in order for users to access the Vault service from their particular network segment?
- How will users engage with you when they need support?

Determining the Scope

Once you have an idea of what users need, determine the scope of the initial phase of onboarding.

Vault is a new service in the sense of how users are intended to interact with it and the sort of features it provides.

Make no mistake, this will be just as big of a learning experience for the support team as it will be for the users.

Adoption Scope - Start Small

With that in mind:

- Start Small: 3-5 Teams/Services
- Start Small: 2-3 Supported Secrets Engines
- Determine the Timeframe

Building the Process

Many would ask why an onboarding process is necessary?

Shouldn't we be using self-service?
For some of your users - specifically those who understand DevOps and automation concepts, this process will be simple.

Many, though, will not meet this criteria. The process is mostly for that group.

Onboarding Process

An onboarding process should look like this:

- Meeting Invite - Contains Service Description, Useful Links
- Users/Group Completes Service Level Requirements Template
- Namespace is Provisioned
- Support Team Meets with Users/Group (Agenda Following)
- Follow-Up Within Two Weeks

IMPORTANT NOTE: Build your process such that users are responsible for managing their own security policy through a version control system. This is especially useful for those who have no DevOps experience, as it provides some exposure to a simple DevOps practice.

Life After Onboarding

Programs to consider to increase adoption:

- Quarterly BrownBag/Lunch and Learn
- Monthly TechTalks
- Internal Vault 101/201 Training

Vault Usage Patterns

User-Based Consumption

- Interaction Through UI/CLI/API
- Authentication Through Directory Service/Third-Party Provider
- Authentication Method Configuration
- Secrets Engine Configuration
- Initial Migration and Rotation of Static Secrets
- Retrieval of Service Account Credentials
- Cloud Credentials
- SSH Key Signing
- SSL Certificates
- Sharing Through Cubbyhole

Service-Based Consumption

- Interaction Through API
- Authentication Through AppRole or Secure/Trusted Introduction
- Rotation of Static Secrets
- Retrieval of Resource Credentials
- Cloud Credentials
- SSL Certificates

Vault Onboarding Summary

- This is a new method of secrets management and requires new methods of ensuring users understand how to consume the service.
- Engage closely throughout the process and invest in users training.
- Ensure that your onboarding process is built such that users are exposed to DevOps practices as part of the onboarding process. T
- This can include something as simple as using Git and pull requests to allow users to manage their own security policy.

Namespaces Overview

Namespaces Overview

Namespaces are isolated environments that *functionally* exist as "virtual" Vaults. They have separate login paths and support creating and managing data isolated to their namespace. This data includes the following:

- Secret Engines
- Auth Methods
- Policies
- Identities (Entities, Groups)
- Tokens

Namespace Functionality

```
$ curl \
  --header "X-Vault-Token: ..." \
   --header "X-Vault-Namespace: ns1" \
  https://127.0.0.1:8200/v1/secret/foo

 vault kv get ns1/secret/foo
```

Namespaces operate as a prefix on the various paths that exist within Vault. Take for example a `KV` secrets engine mounted at `secret/` within the namespace `ns1`:

- Using a relative path (`secret/foo`) while passing a namespace parameter
- Using the fully-qualified path: `ns1/secret/foo`

Policies and Namespaces

```
path "secret/foo" {
    capabilities = ["read"]
}
```

- Any policy rules are *relative* to the namespace in which the policy is defined.
- If defined at the `root` namespace:
  - permits `read` on the fully-qualified path `secret/foo`.
- If defined at the `ns1` namespace:
  - Permits `read` on the fully-qualified path `ns1/secret/foo`

Namespace Considerations

- How is administrative authentication (AuthN) and authorization (AuthZ) structured?
- Do clients that consume secrets operate within a shared platform or within discrete platforms?
- How are cloud accounts structured and managed?
- Are there regulatory or other compliance requrements that need to be asserted and audited by a cenrtal governing body?
- Technical maturity... strong automation capabilities?

Namespace Patterns

1. Do multiple organizational entities share the same authentication method(s)?

2. Does an organizational entity have a requirement for strict, explicit isolation from other entities for regulatory, workflow, or any other reason that may affect the structure of policies, secrets engines, and/or auth methods?

3. Does an organizational entity need to manage their own policies as opposed to having tightly controlled policies managed by a central InfoSec team?

Namespace Component Structure

- Placement of Auth Methods:
  - Interactive (aka administrative) auth (e.g. LDAP/AD) at root namespace for central governance
  - Non-interactive auth (e.g. K8s) at the tenant namespace for consumers of secrets
- Placement of Secrets Engines
  - Mount per namespace?
  - What about root credential rotation? Shared services vs per-team...

Administrative Personas

- **Root Administrator**: audit logging, view/edit vault.conf,
- **Operations Administrator**: responsible for cluster-level operational concerns of a Vault service, such as maintaining the health of a cluster, upgrades, establishing replication relationships between clusters, and managing server and audit logs.
- **Security Administrator**: responsible for cluster-level security concerns of a Vault service, such as managing root-level auth methods and associated policies, creating other administrative-type users, revoking/creating root tokens, and rekey/rotate operations.

Administrative Personas (cont.)

- **Namespace Provisioner**: responsible for provisioning ***top-level*** namespaces and default namespace objects (auth methods, secrets engines, policies, and identity groups & group-aliases).

- **Namespace Administrator**: responsible for managing some or all ***namespace-level*** objects including auth methods, secrets engines, policies, identities, tokens, and child namespaces, depending on the applied namespace management pattern.

Chapter Summary

- Vault Operational Readiness takes preparation
- Need the right people and process
- Understand the separation between Operations, Security, and Consumer
- Your first priority should be an SRE

Reference links

- Life after deployment
- Adopting Vault Guide
- Vault Onboarding: Namespaces and more

Vault Onboarding Users Module Complete!