Vault

Implementation Foundations

Module 5 : Vault Operations

What You Will Learn

Vault Operations

- Overview
- Initialization
- Seal Keys
- Auto-Unseal
- Root Tokens

Logging and Monitoring

- Vault Audit Logs
- Vault Telemetry
- Vault Vault Operations Monitoring
- Vault Systems Monitoring

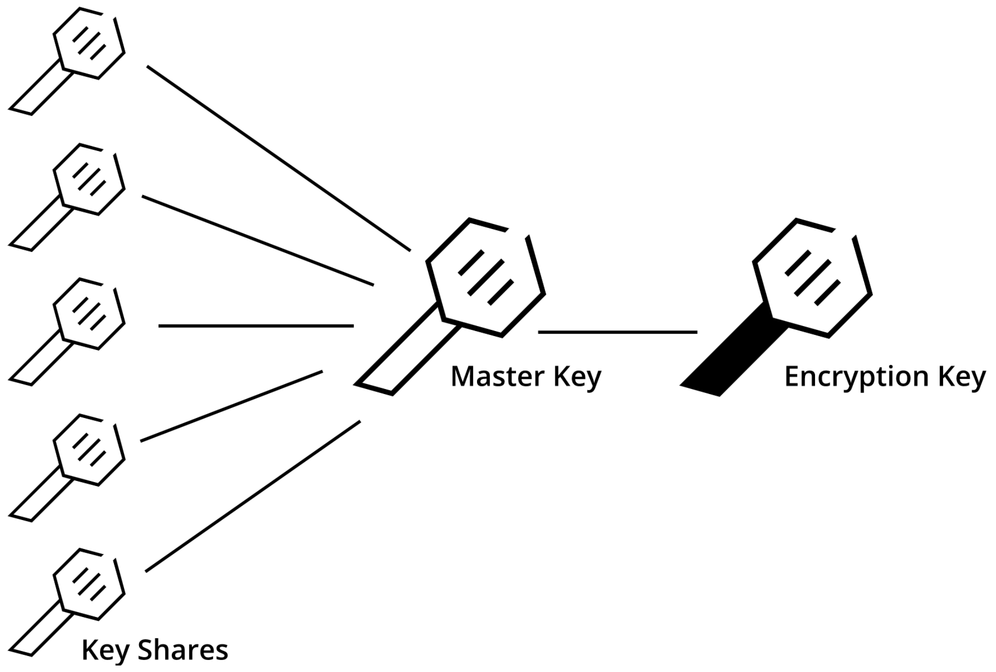
Operations Overview

Introduction to Vault Operations

These initial operational steps are:

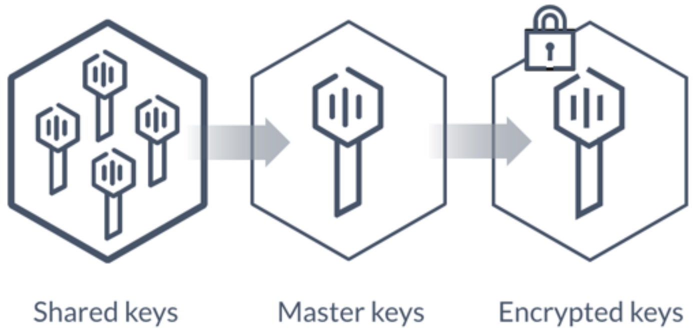
- Initializing Vault
- Management of Seal Keys and Root Tokens
- Configuring Logging and Monitoring of the Vault Service

Seal Keys : Overview



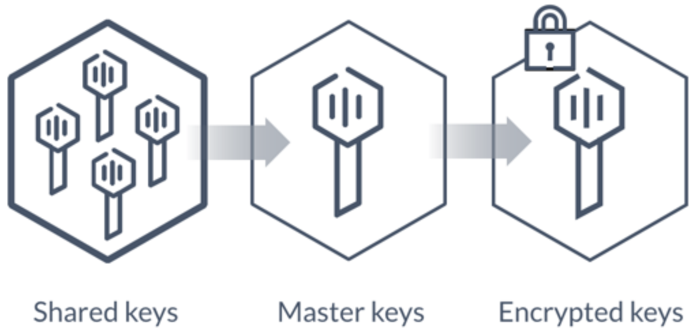
- The first and most critical item created during initialization of Vault is the seal key.
- This is the master cryptography key for to protect the master key.
- If this key is lost and Vault goes into a sealed state, none of Vault's data can be recovered.

Seal Keys : Default Behavior



- During the initialization process, the seal key is "split" into key shares
- This is done using a method known as "Shamir's Secret Sharing.
- This process splits the master key into multiple key "shares"
- Where N number of key shares need to be provided in order to assemble the master key
- This can be define but the default is 5 shares with 3 needed to reassemble the master

Seal Keys : Best Practices



- Seal key shares should be stored in a secure place
- Preferably a Hardware Security Module (HSM)
- Access to seal keys should be tightly controlled, monitored, and audited.
- A rekey is possible and should be preformed anytime the seal keys are exposed

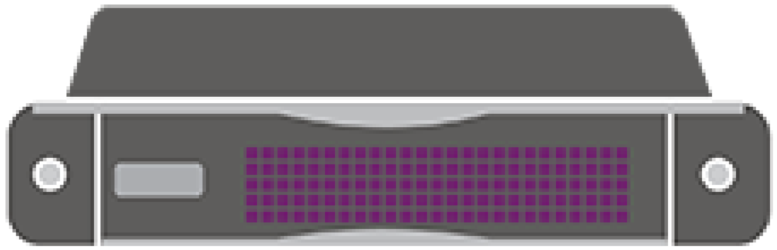
Seal Keys : Recommendations

- Bash history disabled
- Use PGP public keys to encrypt the seal keys
- Rotate seal keys after every use
- Keep a physical copy somewhere
- Remember if the key shares are lost they cannot be recovered

Example Case: Lost Seal Keys

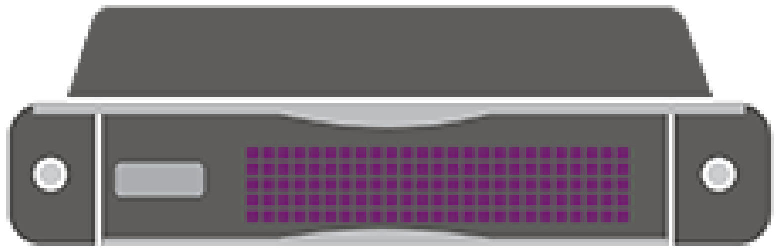
- Large company deploys standalone Vault cluster with SSL certificates and misplaces seal keys.
- Customer-facing services are onboarded to Vault.
- SSL certificates expire, causing Vault to enter a sealed state.
- Secrets are lost, including keys that were used to encrypt all customer data.
- Seal keys were eventually found, but the customer data loss could have resulted in a cost of tens of millions to the company.

Auto-Unseal : Overview



- A way to avoid having to use seal keys frequently is leveraging auto-unseal
- Using an external KMS or HSM to store the seal key
- Minimize the use and exposure of key shares.
- A method of auto-healing after an outage or scheduled downtime.
- Encourages secure storage and management of seal keys

Auto-Unseal : Recovery Keys



- When setting up auto-unseal only one seal key is created and stored in the HSM
- A set of recovery key are created in the same way
- These keys can be used for very high privileged Vault access
  - Rekey of the seal key
  - Perform a root token generation
  - Recovery keys can be protected by a public/private key set

Root Token : Overview

- The second item that is provisioned during initialization of Vault is the root token.
- It gives access to manage and read everything in Vault while Vault is in an unsealed state.
- Note that while Vault is in a sealed state, the root token is essentially useless.

Root Token : Best Practices

- Do not use the root token for anything but break glass situations
- Create appropriate administrative policies, assign them to administrative users, then delete the root token.
- Get in the habit of never needing it. It can always be regenerated if a need ever arises.
- If you determine they would like a root token available, it should be treated in a similar fashion to the seal keys.
- Make sure access to the root token is tightly controlled, monitored, and audited.
- The root token can be protected by a public/private key set

Audit Logging & Health Monitoring

Logging and Monitoring : Overview

Knowing the health or status of your service is key to both making it a great experience for your users and showing value to those above you.

The health of the Vault service can be gathered via

- Audit logs
- System logs
- Telemetry data
- API endpoints.

Vault Audit Logging

The components in Vault that keep detailed information regarding Vault transactions are referred to as "audit devices". There are three primary types of audit devices.

File:
Simply appends audit data to a specified file

Syslog:
Uses the *nix syslog mechanism to write audit data

Socket:
Uses a specified IP address, port, and protocol to write audit data.

Vault Logging : Best Practices

Log Rotate: Ensure log rotation aligns with the speed at which logs are filling up, otherwise, you're gonna have a bad day.

External Logging Services: It is highly recommended that customers employ an external log service, such as Splunk, ELK, or Sumo Logic.

These help to ensure log data is shipped somewhere else immediately to prevent opportunities for log tampering.

Vault Logging : Best Practices (1/3)

Logical Seperation

- Separate log data in a logical fashion
- Gives those teams or application owners access to their own log data
- Allows for self-service for troubleshooting without privileged access

Vault Logging : Best Practices (2/3)

Sane Retention Policies

- This should be set by the business, or preferably by the legal department.
- It should align with other security/legal objectives.

Vault Logging : Best Practices (3/3)

Systems Logs

- It is just as important to monitor system logs, such as /var/log/secure
- This will ensure that the local operating system is being monitored, as well.

Vault Telemetry Endpoints

Telemetry endpoints allow a support team to monitor the health of vault through native integration with the following tools:

- statsite
- statsd
- Circonus
- DataDog
- Prometheus
- StackDriver

Vault Monitoring : Operations

What pieces of information are critical to understanding the health of the Vault service?

- **Number of Service Tokens Created Per (TimePeriod):** This can have an impact on available memory, as well as storage operations in a cluster.
- **Number of Secrets Created:** This can impact available memory, storage operations, and replication.
- **Request Volume per Cluster:** If this is too high, chances are your service is unable to keep up with demand and some user requests are being denied.

Vault Monitoring : Operations (cont.)

- **Seal State:** It's great to be aware of this, regardless of whether auto-unseal is being used, as this directly impacts availability of the service.
- **Replication State:** Is replication keeping up on all the performance replicas? If not, chances are sercret requests are failing.
- **Failed Requests:** Are there requests for certain secrets that are repeatedly failing? This is not just a security concern.
- **Backup/Snapshot Status:** Were the latest data snapshots successful. This is critical for recovery of the service in certain situations.

Vault Monitoring : System OS

What pieces of information are critical to understanding the health of the underlying operating system?

- **CPU:** Although this is rarely an issue in situations where the Vault service is degraded, it should always be monitored.

- **Memory:** Much of the functionality of Vault and its underlying storage are based in memory. Utilization should always be closely monitored.

Vault Monitoring : System Disk

- **Disk Space:** Vault doesn't have a huge storage footprint, but this should always be monitored in the rare situation that it does.

- **Disk Inodes:** Because Vault is storing lots of tiny pieces of data, and because customers may not use large disk for Vault clusters, exhausting inodes is a possibility and can have a similar impact to running out of disk space.

- **Disk I/O:** The performance of the underlying storage for a Vault cluster can impact things such as creation of service tokens, or even replication performance.

Vault Monitoring : Network

- **File Descriptors:** File descriptors are opened in Linux for various reasons, the most common being open network connections. If a system is approaching limits on file descriptors, chances are network connections are filling up.

- **Network Throughput:** Depending on how heavily utilized a cluster is and how many other clusters it may be replicating to, bandwidth may be an issue.

Vault Monitoring : Security

What pieces of information are critical to understanding the security posture of the Vault service?

- **Root Token Use:** This should only ever exist in situations where a break glass token is needed. In those cases, it should only ever been used in very rare situations.
- **Root Token Creation/Rotation:** This may be occurring through standard process/protocol, but is still a notable security event.
- **Seal Keys Rekey:** This may be occurring through standard process/protocol, but is still a notable security event.

Vault Monitoring : Security (cont.)

- **Repeated Authentication Failures:** The random authentication failure should be expected. If the failure is happening repeatedly in a short period of time could potentially indicate security issues.
- **Repeated Access Failures:** Something repeatedly failing to access a certain resource isn't necessarily a security issue, but should be considered suspicious.
- **Tokens Created with Long TTL:** Ensure the TTL for any given token doesn't violate the policies around credential lifetime.
- **Vault/Secret Accessed from New IP:** Some tools, such as Splunk, are capable of tracking deviations. This includes a secret being accessed from a new IP or subnet.

Vault Operations : API Endpoints

- Operations endpoints are located under the /sys/ path in the API.
- Some of the more useful API operations for monitoring the health of the service are listed here:
    - **sys/audit:** List, enable, and disable audit devices. If all audit devices become unavailable, Vault will hang and become difficult to recover.
    - **sys/health:** A summary of the health status. Great for determining the state of a cluster.
    - **sys/host-info:** Displays hardware utilization data for a given host.
    - **sys/leader:** Displays the HA status and current leader of a cluster.

Vault Operations : API Endpoints (cont.)

- **sys/metrics:** Displays various telemetry metrics for Vault.
- **sys/replication/performance:** Displays the current status of performance replication.
- **sys/replication/dr:** Displays the current status of DR replication.
- **sys/seal-status:** Displays the seal status of Vault.

Chapter Summary

- Introduced the first couple of commands you'll run with Vault and best practices around those initial secrets
- Overview of the day to day concerns and operational concerns of running Vault
- Discussed in depth the API endpoints and functional endpoints you'll need running Vault

Reference links

- Vault Operator Commands
- Vault Telemetry
- Vault HTTP API Reference for System Backend

Vault Operations Module Complete!