HashiCorp Vault

Unlocking the Cloud Operating Model: Security

Contents

Understanding the security benefits that come with managing Identity and Secrets in a multi-cloud environment....03

What is Identity-based Security....04

Challenges with Multi-Cloud Secrets Management....05

HashiCorp Vault: Multi-Cloud Secrets Management Simplified....06

Adoption, Operationalizing, and Scaling with Vault....07

Summary....08

Understanding the security benefits that come with managing Identity and Secrets in a multi-cloud environment

For many enterprises, the logistical realities of cloud adoption often requires a shift from a traditional on-premise static infrastructure with clearly defined network perimeters, to cloud infrastructure, that is highly dynamic and has no clear network perimeters.

This shift in operating models requires a fundamentally different approach to security: instead of focusing on a secure network perimeter with the assumption of trust, the focus is to acknowledge that the network in the cloud is inherently "low trust" and move to the idea of securing infrastructure and application services themselves through a trusted source of identity and secrets management.

In this white paper, we look at the implications of the cloud operating model, and present solutions for IT teams to adopt this model at the security layer.

What is Identity-based Security

Managing access to secrets in a multi-cloud world consists of two primary user types: humans and machines. Managing secrets for humans can be a fairly straightforward experience. Systems simply need to permit or restrict access to create or manage secrets, or manage others who may have access, based on the identity (or identities) the user logged in with.

Machines, however, are a different issue, as they can include servers, virtual machines, applications, microservices, scripts, and more, all potentially needing access to different systems and secrets. When it comes to managing secrets with machines in a multi-cloud environment, the dynamic nature of HashiCorp Vault comes to the forefront.

The ability to create dynamic secrets per service has several benefits, secrets can be short lived, specific secrets can be revoked in the event of a breach, and all actions are audited per secret. Each cloud service can be given access to secrets based on their identity and have a policy associated with it. With Vault, whether a user is looking to create and distribute organizational secrets and access or applications are looking to retrieve new database credentials every 15 minutes, centrally managing this access based on trusted identities is critical.

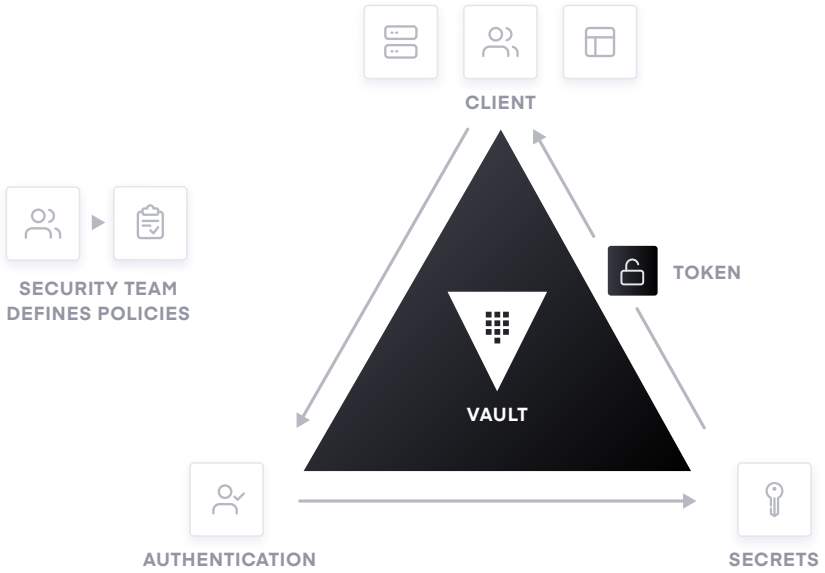

Figure 1: Operating Model for Identity-based Security and Secrets Management

Challenges with Multi-Cloud Secrets Management

Dynamic cloud infrastructure means a shift from traditional, statically defined, host-based identity to application-based identity in an environment with low-trust networks across multiple clouds and no clear network perimeter. In the traditional security world, we assumed networks were internal and inherently high trust, which resulted in a hard shell and soft interior. With the modern "Zero Trust" network approach, we work to harden from the inside as well. This requires that applications are explicitly authenticated and authorized to fetch secrets and perform sensitive operations, all while being tightly audited.

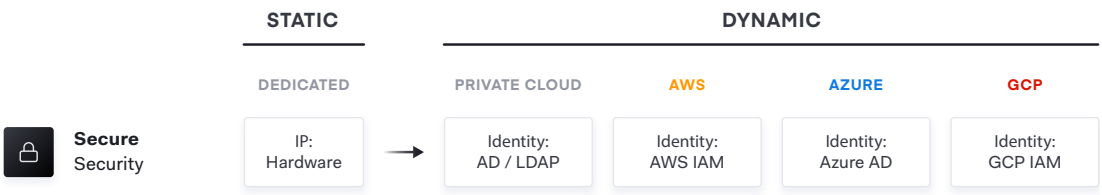

Figure 2: Static to Dynamic infrastructure at the secrets management layer

We talk to organizations of all sizes about their infrastructure plans and adoption of the cloud operating model as they navigate their transition to the cloud. As this migration is taking place, some organizations can struggle with the discovery of a sprawl of secrets and access credentials sprinkled throughout their infrastructure.

Here are a few examples of what this secrets sprawl can look like:

- Hardcoded hostnames or firewall rules used for application identity
- Plain-text usernames/passwords embedded in scripts, configuration files, and source code
- Highly privileged cloud provider API keys in source code
- Certificates and encryption keys stored on the filesystem unencrypted
- Staff can be hesitant to change access because they are not sure what will break
- Limited audit logging on who is doing what and where

We recommend the adoption of a secrets management solution for the handling of secrets and access credentials during this transition to the cloud, as these types of risky problems can be amplified when the secrets sprawl extends out beyond trusted inner networks. When we talk about secret management, the goal is to solve the lack of visibility and control around the handling of these highly trusted credentials.

HashiCorp Vault: Multi-Cloud Secrets Management Simplified

HashiCorp's Vault enables teams to securely store and tightly control access to tokens, passwords, certificates, and encryption keys for protecting machines, applications, and sensitive data. Vault allows you to centrally manage and securely store secrets across on-premises infrastructure and the cloud using a single system.

The Vault API exposes cryptographic operations for developers to secure sensitive data without exposing encryption keys. Vault also can act as a certificate authority, to provide dynamic short-lived certificates to secure communications with SSL/TLS. Lastly, Vault enables brokering identities between different platforms, such as Active Directory, AWS IAM, and LDAP into unified identities to allow applications to work across platform boundaries.

Vault, running as a centralized service, enables IT teams and organizations to provide secrets management and data encryption services across large fleets of applications and engineering teams, all while globally managing policies and delivering consistent security through a single workflow.

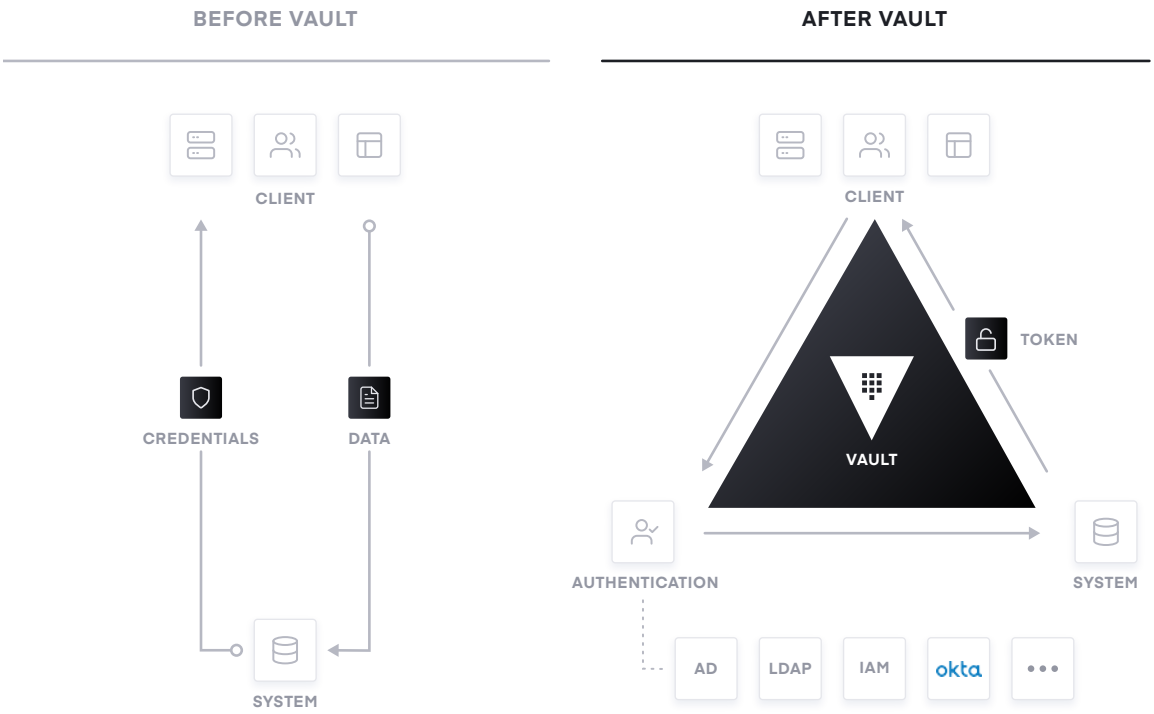

Figure 3: From traditional secrets management to modern service networking with Vault

Adoption, Operationalizing, and Scaling with Vault

Adoption of Vault often follows a three-stage rollout pattern for most organizations: Adoption, Operationalizing, and Scaling. Each stage gradually adds more automation and operationalization to the process and more software-driven technologies to improve agility, performance, and security at a reduced cost. Your organization can use these steps as a playbook for guiding the transition to the cloud and assisting with many of the challenges your organization will face.

Adoption: Centrally Managing Secrets

The first stage of Vault adoption is securing secrets through a centralized location, eliminating secret sprawl that exists today and properly managing who can access what, when did they access it, etc. By centralizing secrets and introducing proper access controls, development, operations, and security teams can take a huge step towards securing applications and sensitive data along with strengthening their security posture. Vault uses policies to codify how applications and users can authenticate, which secrets and operations they are authorized to use, and how auditing should be performed. Vault integrates with an array of trusted identity providers such as AWS, Azure, Google Cloud, Alibaba Cloud, Kubernetes, Active Directory, Okta, and other SAML-based systems for authentication. Vault authenticates these identities and uses them as a system of record to manage and enforce access to secrets and systems.

Operationalizing: Application Onboarding & Legacy Tokens

In the Adoption stage, secrets and access have been centrally managed within Vault. Now, how do you make consumption of those secrets easier? This often happens through your cloud's orchestration tools. Can your orchestration tools actually facilitate the secure introduction of those secrets to the underlying consumer applications?

For example, maybe you have a VM and you are using a tool like Terraform to securely introduce secrets for your applications that are not aware of Vault. Or, perhaps it's a Kubernetes cluster and you want to inject those secrets into the different pods that may be consuming them (this can be achieved via sidecar process to inject secrets into the file system or environment variables).

Streamlining the lifecycle of secrets and making them easier to consume through various strategies means you don't have to rewrite all of your legacy applications you are transitioning into the cloud. Additionally, you have provided an easy way for greenfield applications that may be running on newer orchestration platforms, such as Kubernetes, to be able to consume secrets in a safe manner.

Scaling: Dynamic Secrets & Encryption as a Service

As we finish making secrets easily consumable, how can we shorten the time in which an exposed secret can be used? Can I, as a requester of a secret, get a different secret every time I make that request? Can the secret have limited time-to-live access? You can do this with dynamic secrets in Vault. This ensures that if the requester leaves the company, or a container gets moved to a different host, the secret becomes invalid, and I don't have to have a human that's responsible for remembering to go revoke that secret. Imagine an attacker gets access to a system and the secrets on disk. With dynamic secrets, this attack surface can be mitigated with single-use or short TTL. Now multiply these scenarios across an entire organization that could potentially have thousands of secrets. Vault can be a central hub to issue and revoke all of them.

Additionally, organizations need to protect application data at rest and in transit (especially in a cloud environment). Vault can provide encryption as a service as a consistent API for key management and cryptography. This allows developers to interact with a single endpoint to protect data across multiple environments. Using Vault as a basis for encryption solves difficult problems faced by security teams such as certificate and key rotation. Vault enables centralized key management to simplify encrypting data in transit and at rest across clouds and datacenters. This helps reduce costs around expensive Hardware Security Modules (HSM) and increases productivity with consistent security workflows and cryptographic standards across the organization.

Summary

The move to a cloud operating model involves a shift in thinking and the tooling we use to secure our infrastructure. Traditionally, we had a relatively static world of dedicated servers, static IP addresses, and a clear network perimeter. However, in the cloud we have ephemeral and elastic pools of infrastructure with dynamic IP addresses, and no clear perimeter.

The adoption of this operating model for identity-based security and secrets management with Vault is an inevitable shift for enterprises aiming to rationalize security in a world of multiple clouds.

If you would like to learn more around Vault, there is a great Introduction to HashiCorp Vault with Armon Dadgar, which helps to quickly explain Vault from the ground up.

Try Vault by getting hands-on via our Learn Portal.


HashiCorp