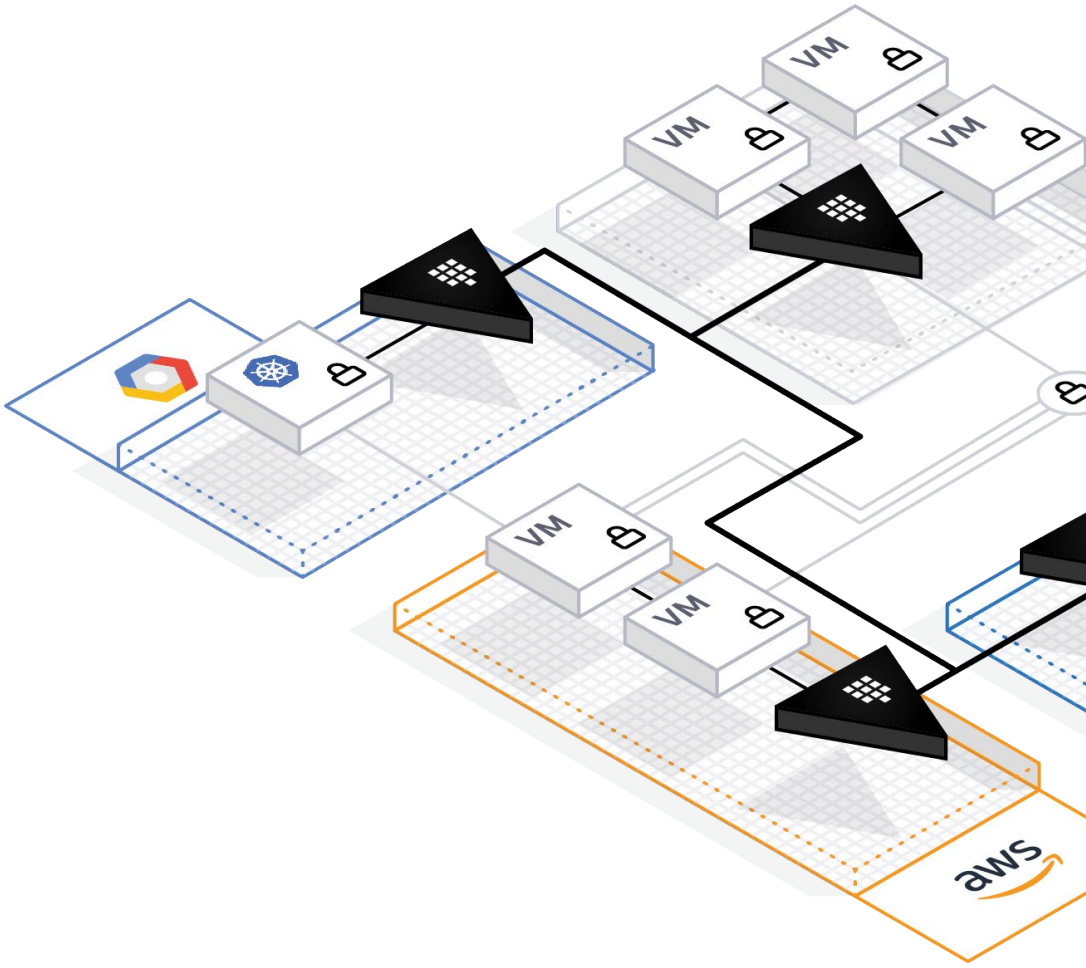
HashiCorp Vault

Top Security Challenges in Digital Transformation



WHITEPAPER

HashiCorp

Contents

Introduction

As we described in a previous white paper on Cloud Adoption, moving from traditional development and deployment to a modern, cloud platform presents a unique challenge. If you go from an on-premises, waterfall deployment basis to a modern, cloud-based world, you need to decouple your applications from your infrastructure.

The primary challenge is heterogeneity: How can operations, security, and development teams apply a consistent approach to securing this infrastructure—and do it efficiently?

Currently, most organizations address this challenge with ad-hoc security practices, custom encryption, and secrets management that varies by platform. Heavily regulated industries, such as finance and healthcare, have the added complexity of auditing and maintaining industry practices that are defined...

- by the industry: PCI-DSS, HIPAA, GLBA, SOX, etc.
- by local or federal government: FDA Title II, FIPS 140, DPA, etc.
- by supra-governmental entities: GDPR, etc..

Multinational organizations—or those with multinational customers—often need to follow a range of regulations, some of which overlap.

The Challenge

In a traditional data center, regulatory requirements are addressed by each individual business unit, potentially via a central security department that provides governance on compliance requirements. This means implementation and enforcement is federated to individual teams.

- privileged access management
- static secrets management—e.g., third-party access keys
- database credentials—generally provisioned manually, rotated rarely, and stored with the application
- PKI—both client and server certificates
- encryption keys—commonly stored with the application

In these environments, policy is enforced by operators and relies on the operator's ability to quickly delegate trust, identify workloads, and provide the right level of access. Policy may also potentially be translated into individual RBAC models from different tools.

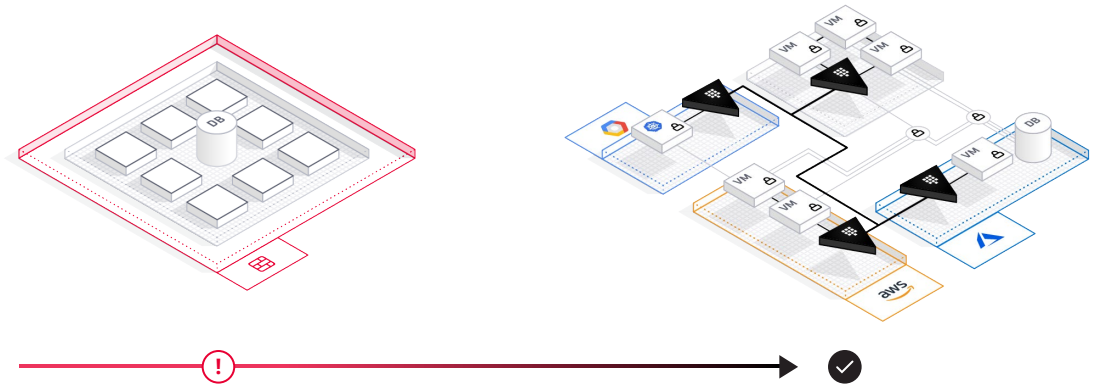
In the past, this model has served organizations well from a security perspective, but it's been one of the factors that have slowed innovation.

Security should be implemented by design, rather than as an afterthought delegated to third-party software. This is especially true where policy is expressed in silos—across multiple software solutions.

Organizations encounter these issues when going through digital transformations—specifically around one or more of the following four initiatives:

1. Cloud Adoption

Traditional datacenter security is mostly focused on securing the perimeters around workloads and users. Common concepts include PCI zones—where data inside a perimeter is considered secured and limitations are set at the physical level or equivalent alternative.[1] But this becomes particularly challenging when moving to cloud providers that don't share private networks.



Static Infrastructure

Datacenters with inherently high-trust networks with clear network perimeters.

Dynamic Infrastructure

Multiple clouds and private datacenters without a clear network perimeter.

Such a physical perimeter doesn't exist in a public or hybrid cloud. Instead, the separation is established with logical access limitations, software-defined network policies, and encryption.

2. Microservices Architecture

The data flow within secure applications has traditionally been constrained to monolithic applications. Today, the most common design pattern involves using API contracts between microservices, which limits the potential "blast radius" and accelerates development.

These modern designs are generally provisioned and scaled through various degrees of automation. However, such automation requires a programmatic way to establish identity—to provide the right level of access for the workload.

1 "Information Supplement - PCI DSS Cloud Computing Guidelines" - PCI DSS v2 - Section 4.4.3 "Segmentation Technologies"

Because of the ephemeral nature of the workloads, you can't rely on manual provisioning of secrets (e.g., API keys, TLS certificates, database credentials).

Traditional security solutions—either based on enterprise software or hardware security modules—don't scale to a large number of operations per second, so developers and operators tend to revert to bad practices, such as storing secrets in clear text or using static credentials.

3. DevOps Adoption

In addition to the points above, the security policy should be treated just the same as any application code:

- defined as code
- versioned as code
- tested as code

As infrastructure is modeled and industrialized—through configuration management or other patterns—systems require secrets to be available at runtime, automatically, and as fast as possible.

4. Agile Development

Development teams need to focus solely on delivering value to the business—and security is no exception. Abstracting underlying operations is paramount to the teams' effectiveness.

Teams should be able to simply consume encryption as a service, with key rotation and key strength being managed by security experts. Teams should also have simple access to test infrastructure—with not only the right level of access, but also the right credentials to deploy applications in different environments.

Ultimately, from a process perspective, different workloads should have unique and programmatic processes to consume secrets throughout the whole development process (whether those secrets are certificates, encryption keys, credentials, or API keys).

The Key Elements of Effective Data Security

In this new wave of digital transformation, a centralized security solution must be able to establish identity for a wide range of consumers—including humans and workloads.

Modern workloads are continually being provisioned for a number of environments—testing, development, QA, production. In this world of continuous integration pipelines or other orchestrated process, identity and access to secrets must be established programmatically. In the same way, developers, operators, DBAs and other groups within an organization must be able to perform CRUD operations on secrets.

It's crucial that all this happens in a user-friendly way, based on roles in an existing identity provider. Also, both users and machines should be subject to a centralized policy and a single point of audit.

Short-lived secrets, generated on demand, are better than traditional, cumbersome password-rotation practices. Using secrets requested programmatically and with a set expiration is more scalable and less error-prone.

Consuming cryptographic services—e.g., encryption/decryption, signing/verification, HMAC—in a way that abstracts complexity away from developers is key to "shifting security to the left" (i.e., designing it in from the start). This abstraction of complexity ensures that security is built into the application without the need to maintain cryptographic functions.

Cloud-Native, Centralized Secrets and Privileged Access Management

Most existing software is designed around traditional data centers and monolithic applications. Hence, most of these products require a considerable amount of data-center resources—highly available storage and network, hardware security modules, even external HA software, just to name a few.

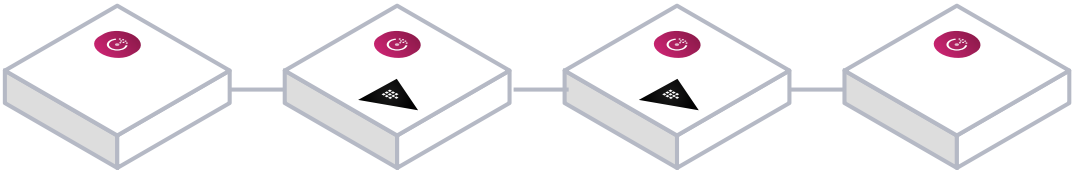
It's typically built with the traditional "Castle and Moat" approach, where secure data only exists within a network segment and regulatory requirements translate into a new "castle," thus increasing operational overhead.

In addition, the sheer growth of IT demands that the cryptographic performance of security software scales just like "normal" APIs. Scaling this way allows programmatic access to secrets and security services, just like any other service in modern IT.

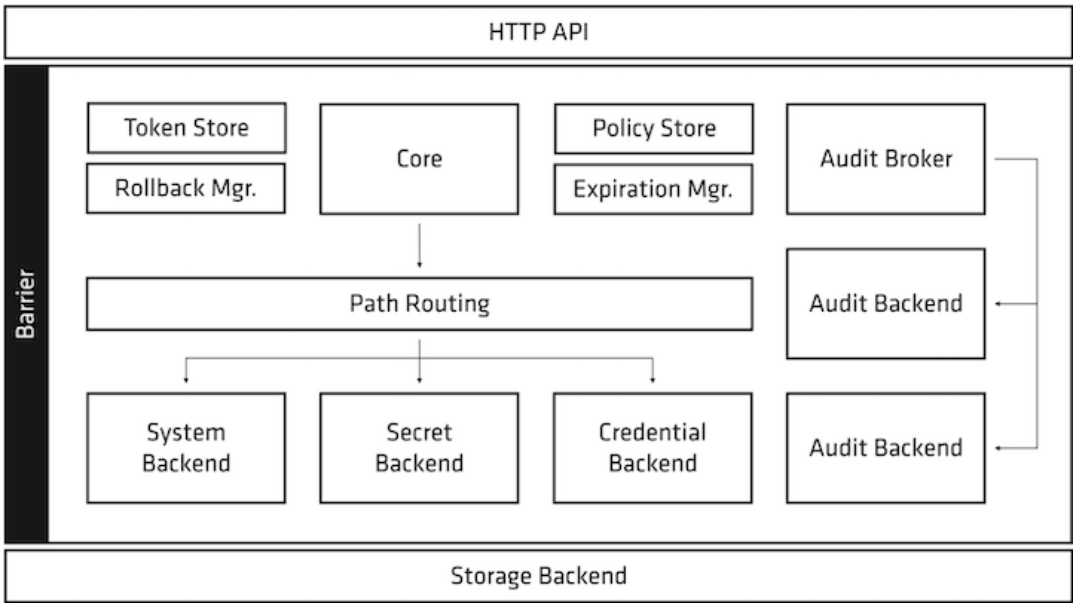
HashiCorp Vault is designed with all these considerations in mind. One of its significant advantages is that it's under the constant scrutiny of a large open-source community of users, developers, security experts, and even independent third-party security auditors.

Unlike traditional security solutions, Vault embraces the open-source community to create a more secure, robust and scalable product. It supports a broad spectrum of use cases, driven mostly by customer demand.

Vault makes minimal assumptions about hardware configuration, and is designed for cloud environments. It is natively highly available, with the advantage of persistence from HashiCorp Consul's key-value store.



Vault can delegate trust to a hardware security module, or a platform-specific product, such as AWS KMS, Azure Key Vault, or Google Cloud KMS. But by assuming there is no secure perimeter or cryptographic root of trust, Vault introduces the concept of a cryptographic barrier—where all data at rest is encrypted using a AES-GCM storage key. This key is itself encrypted by a master key, and can be rotated online on demand. Finally, everything sent to clients is encrypted before transport using TLS.



The master key is the root of Vault's cryptographic model and can be secured through different methods:

- Where trust is delegated to a number of individuals in the organization, the master key can be reconstructed using Shamir's secret sharing, which requires multiple key holders to enable active operation and maintains a separation of privileges.

- For environments that require a high level of automation, trust can be delegated to a cloud platform's key-management solution. In this scenario, Vault's master key is encrypted by KMS or Key Vault and automatically unencrypted on initialization.

- For highly regulated environments, the same model can be applied using an HSM as the root of trust.

Similar trust principles apply to doing a secure introduction (introducing the first secret to a machine). Machine workloads can be identified using a trusted entity, such as:

- AWS IAM
- Google oAuth
- Azure Resource Manager
- Kubernetes
- HashiCorp Nomad
- TLS Certificates
- AppRole, for generic orchestration methods including configuration management

HashiCorp Vault can store or dynamically generate a wide variety of secrets. Based on the principle of least access, a user or workload can interact programmatically with Vault—via its API, CLI or UI—to produce and consume secrets using:

- A key-value store, segmented by logical "mounts"
- A self-service certificate authority (if permitted by identity and policy)
- A multi-cloud platform credential generator
- An SSH dynamic key/password/CA
- A database access management solution, creating dynamic, short lived credentials based on specific roles, and aligned to different "grants"
- Encryption operations "as a service," with pre-provisioned keys that allow users or workloads to encrypt/decrypt, sign/verify, HMAC, and even obtain entropy

Vault performs all of this under a single rich policy that provides both per-secret and per-request governance, as well as a single point of audit.

HashiCorp Vault can also map identities from multiple access systems per user. It can merge access permissions from those systems, and group them into a single persona. This allows users to be subject to a single policy, regardless of authentication method.

Conclusion

Over the last 50 years, the IT industry has undergone a number of transformations—from mainframe monolithic systems, to commodity hardware, to virtualization.

At HashiCorp, we believe the industry is undergoing a new transformation into utility compute. This transformation requires organizations to adopt more dynamic development and operational practices—such as Agile and DevOps.

With each transformation, new security challenges arose, and existing solutions required overhauling. Right now, end-to-end automation demands performance and scalability from security systems. Systems are expected to be resilient and self-healing, with minimal operational overhead.

We are certain that no one will opt to compromise on these aspects of application development.

As the software-defined data center becomes the new paradigm, organizations that want to remain competitive and lead innovation in their respective markets will undergo a digital transformation. HashiCorp Vault is the defining security solution for these leading organizations.


HashiCorp