HashiCorp

WHITE PAPER

12 things a modern secrets management solution must do

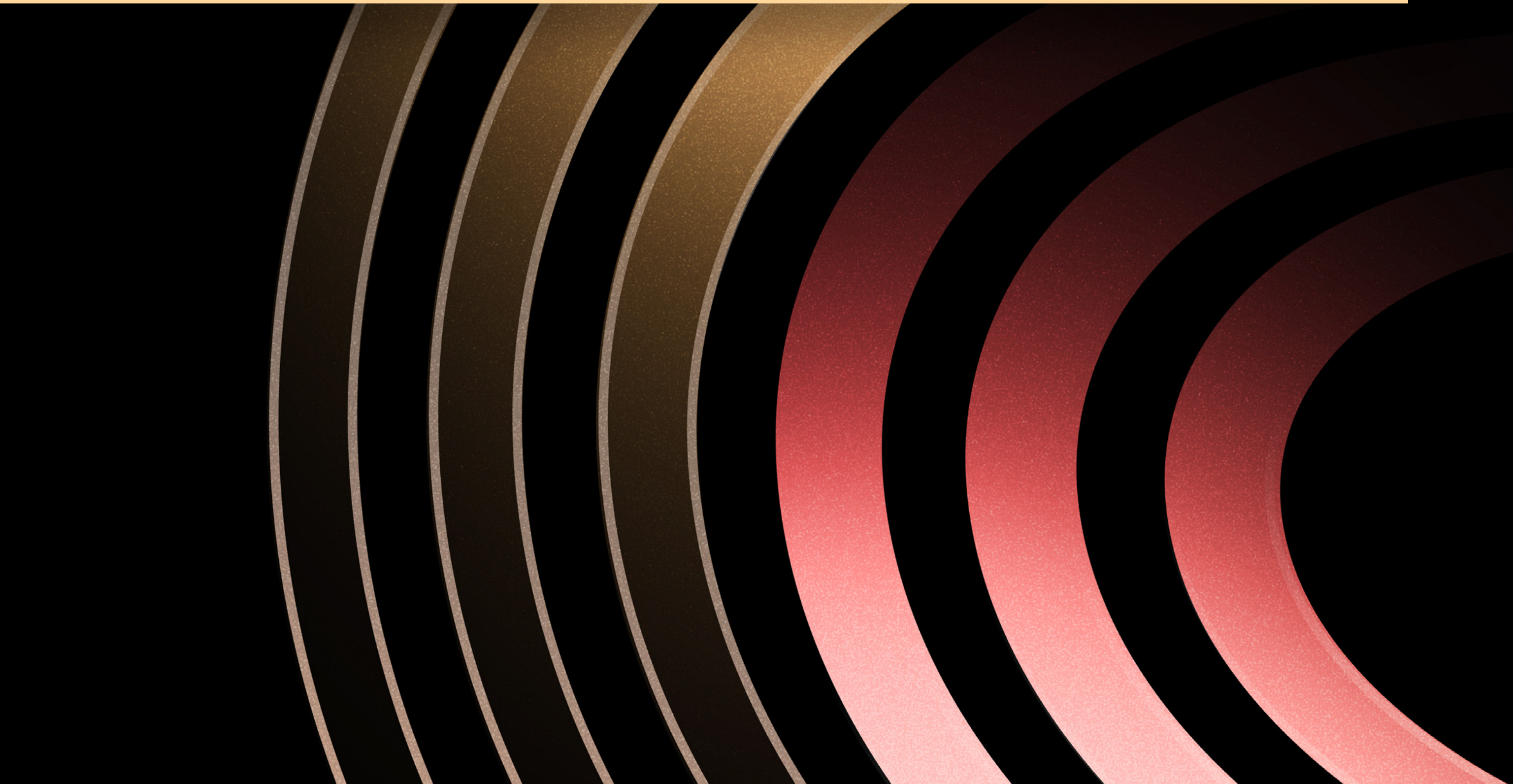


Table of Contents

Executive summary

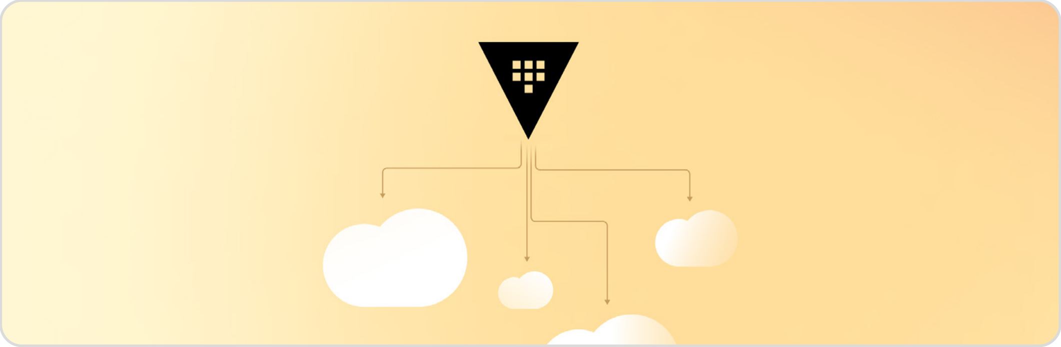
Keeping secrets is a critical but unsung component of cloud computing. Properly managing, monitoring, and auditing secrets in a single location is a fundamental requirement for managing modern infrastructure. But as organizations adopt distributed, multi-cloud infrastructures, they generate far more secrets (sensitive information such as passwords, tokens, database connection strings, cryptographic API keys, etc.) than ever before.

They must also modernize how they manage and provision secrets to avoid secrets leakage, secrets sprawl, and other costly, even dangerous, secrets management mistakes. That's why secrets management was cited as important or very important to cloud success by three-quarters of the respondents to the 2023 HashiCorp State of Cloud Strategy Survey.

Modern secrets management solutions help organizations more precisely control how secrets are stored, tracked, transmitted, accessed, rotated, and revoked — no matter where they are in the organization's digital estate. So to meet the needs of today's organizations, a secrets management system has to perform a wide variety of key functions:

Policy-based access controls and authentication methods are needed to enforce access privileges for people, devices, data, and applications. Support for dynamic secrets, generated on demand and revoked when no longer needed, is essential to limit the damage of leaked secrets. A modern secrets management system should use APIs to access and manage all secrets — even those held on other platforms. Secrets versioning lets DevOps teams roll back secrets that expose security vulnerabilities or impede access to networks and applications, while robust auditing and monitoring capabilities are needed to track who or what accessed which secrets.

This white paper details the twelve essential capabilities of a modern secrets management solution and why a robust, platform-agnostic secrets management solution is vital to success in a multi-cloud world of distributed applications, services, and infrastructure.



Why secrets management matters

Protecting secrets is now exponentially harder and more critical than it was before the onset of the cloud. In the days of on-premises datacenters, admins had to manage a relatively small collection of personal and group application passwords — all stashed "safely" in the datacenter behind the firewall. But with the rise of distributed applications, short-lived remote connections, and hybrid multi-cloud infrastructure, organizations are accumulating and using far more secrets than ever before. One large healthcare organization, for example, processes 300 million requests for secrets *daily*.

Of course, modern secrets include much more than simple passwords, including a wide variety of sensitive information such as tokens, database connection strings, cryptographic API keys, and more. To ensure those secrets are both protected and available for use when needed, organizations increasingly rely on secrets management platforms to provide secure access to applications, databases, networks, and infrastructure. These platforms manage the distribution and provisioning of secrets while automating such fundamental practices as secret rotation and revocation, as well as compliance reporting.

Secrets management helps you control how secrets are stored and transmitted, when they're used, how frequently they're rotated, and how easily they're revoked — across your entire digital estate. The practice and technology of secrets management has become more important and more rigorous to keep pace with the increasing complexity of enterprise infrastructure, the corresponding expansion of attack surfaces, and the rise in breaches. Once mainly the concern of software engineers and security professionals, secrets management has been elevated to the front lines of an identity-based, zero trust security strategy where every request must be authenticated and authorized and network and application access is allowed according to the **principle of least privilege.**

Secret leakage is a widespread problem

Organizations are upgrading their secrets management strategies because they have little choice. Leaked secrets, such as passwords, personally identifiable information (PII), and credentials, such as authentication data for identity verification, are now the most commonly cited security threat, according to the **2023 HashiCorp State of Cloud Strategy Survey**. And a **2023 Verizon report** on data breach investigations stated that 37% of firms have experienced credentials-related data compromise incidents, and 45% have dealt with incidents involving stolen credentials.

Credential loss can occur when organizations eschew secrets management best practices. In its **State of Cloud Security** report, Datadog notes that, "Long-lived credentials continue to be a risk," as too often, "unused access keys are still not being deprovisioned." For example, nearly half (49%) of respondents said they had a cloud provider access key that had not been used in the past 90 days, up from 40% a year ago. And one in three had active credentials older than 1 year that have not been used in the past 30 days, up from 25% a year ago.

Writing about stolen credentials-related attacks, Verizon's report notes, "Poorly picked and protected passwords continue to be one of the major sources of breaches." At the same time, the cost of mitigating data breaches and plugging security gaps continues to rise: **IBM calculates** that the average price of a data breach hit $4.45 million in 2023, up 15% in the last three years.

The rise of dynamic secrets

With fundamental changes to enterprise infrastructure and security brought on by the cloud, remote access, and other wide-ranging forms of **business disruption**, few organizations can sustain legacy, manual secrets management approaches. Manual secrets management is challenging even for smaller organizations, virtually impossible to scale, and increases the risk of a breach caused by unsecured and untracked static secrets.

That's why one element of a modern secrets management strategy focuses on the shift from static to dynamic secrets. Static secrets, such as hard-coded credentials, raise the risk of security breaches because they are typically predefined ahead of time, are often shared, and may be stored in plaintext where attackers can easily access and misuse them.

On the other hand, dynamic secrets are generated on demand and are unique to a single client (human or machine) and access request. A dynamic secret exists for a limited time, or for a single interaction, and is revoked when it expires, which drastically limits the time cyber attackers can use them if stolen.

Embracing an automated zero trust security model

Until the advent of multi-cloud architectures and the need to support ephemeral remote access, solutions for safeguarding infrastructure, data, and access were typically rooted in securing network perimeters based on the IP address of the entity requesting access. Applications talking to databases, users accessing hosts and services, servers talking to each other, and other actions were allowed or restricted based on whether the IP address was located inside or outside the firewall. This is the classic "castle-and-moat" approach to security, where every request for resources beyond the network "moat" was deemed unsafe, while everything inside the proverbial castle walls was considered trustworthy.

But like a castle-and-moat, the IP approach hasn't stood the test of time and evolving technology as the world moved beyond monolithic, on-premises infrastructure. Operating in the cloud escalates operational complexity: instead of being fixed, IP addresses are dynamic and ephemeral as newly spun-up servers need to talk to each other across clouds, while mobile and hybrid workers must be able to access shared resources no matter where they are.

Embracing today's multi-cloud world of dynamic, ephemeral infrastructure requires organizations to modernize their secrets management approach. The solution is to apply zero trust practices and on-demand automated provisioning, auditing, and multifactor authentication for access.

In a zero trust model, all individuals, devices, and services that seek organizational resources face continual verification, no matter where they originated, whether behind the firewall or from third-party cloud and network services. That's why a modern secrets management solution is an essential component of a robust enterprise zero trust security strategy.

Fighting secrets sprawl with centralization

When administrators manage secrets for apps in multiple locations without using a centralized solution, they may have little understanding of where their secrets are stored. Secrets sprawl — secrets stored in multiple places, sometimes hardcoded into configuration files, Git repositories, and many other locations — means SecOps teams may not even be aware of all the secrets they need to protect. That makes secrets incredibly difficult to secure and offers few hints on how to respond to a breach. Coping with secrets sprawl requires a secrets management platform to manage secrets from a centralized location.

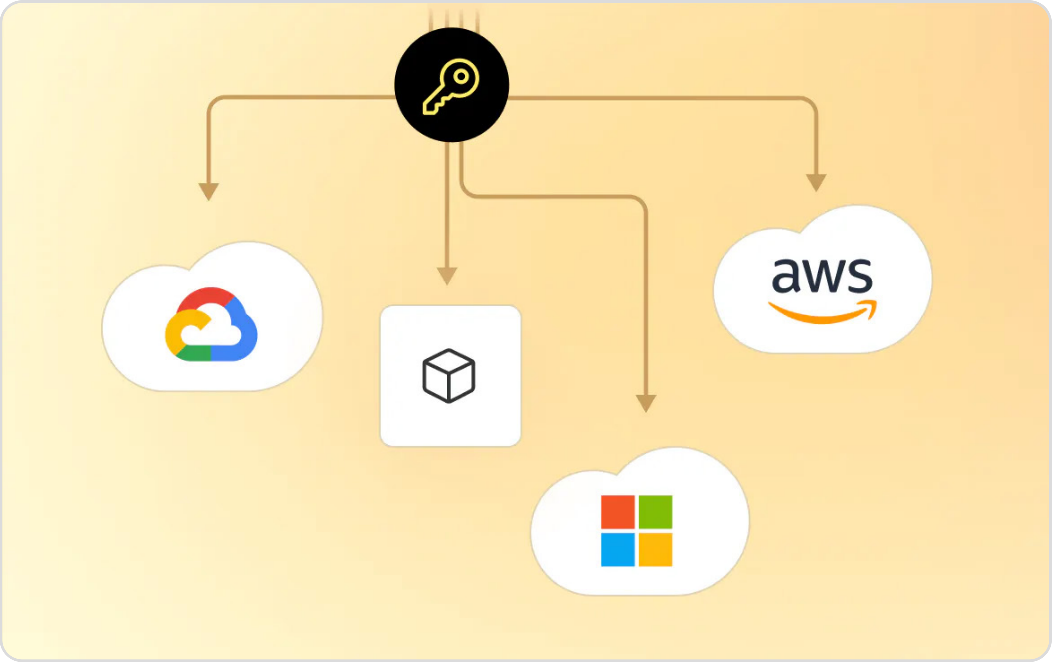
Managing secrets centrally enables well-coordinated rotation and revocation of secrets across multiple cloud platforms, services, and applications. Centralization not only speeds up the process but also reduces the rate of human error and improves auditing for compliance purposes. SecOps teams favor a centralized approach because it enables them to establish best practices founded on acknowledged security principles. Best of all, perhaps, centralization of secrets management lets SecOps teams do their work without having to involve developers.

Secrets management deployments have become more pervasive, but some companies still write their own or tap off-the-shelf solutions from their cloud service providers — options that can prove limiting. It can be expensive to continually scale and update homegrown secrets management solutions, with no assurance of achieving state-of-the-art functionality. And relying on a cloud service provider's proprietary platform for administering secrets may not cover your entire digital estate and could require you to learn and manage multiple secrets management solutions.

For example, GitHub, a popular software development and version control platform, created a solution that required manual custom configurations and code to connect hundreds of apps and services while managing the keys and secrets by hand. Any changes to the static secrets protecting the company's services and applications meant changing and updating that secret in connected systems.

"In the past, we'd used a range of homegrown systems for everything from secrets management to load balancing that required a ton of custom integration and custom coding across hundreds of services and thousands of nodes," explained Scott Sanders, GitHub's Vice President of Infrastructure, in a HashiCorp case study. "But as we started adding more applications, more nodes, and more users, that model became untenable. We needed a more efficient, standard, and automated way to support a dynamic and growing user base."

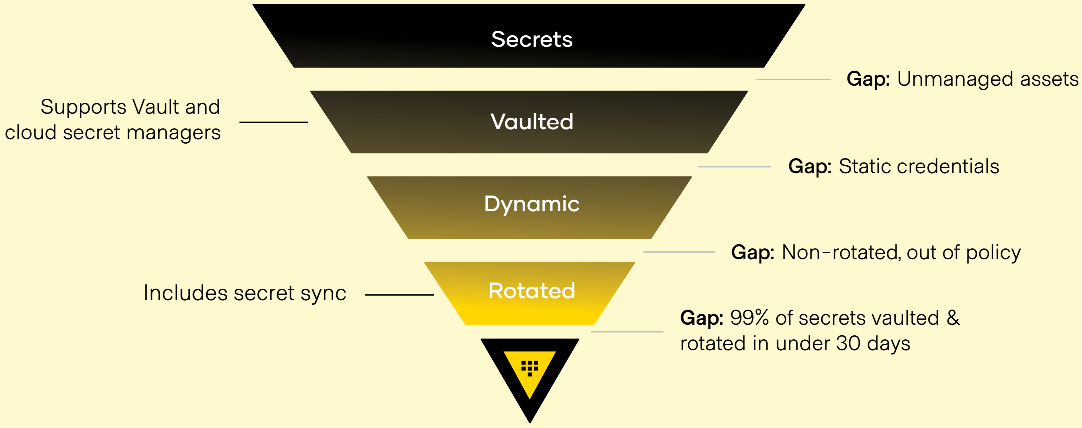
To ensure your organization gets what it needs, look for these capabilities in your secrets management



The 7 stages of the secrets management lifecycle

There are seven stages of the secrets lifecycle, and a modern secrets management solution must address all of them:

1. **Creation:** The initial stage where secrets are generated or created.
2. **Storage:** Once secrets are created, they must be securely stored to protect them from unauthorized access.
3. **Access control:** Only authorized users/ systems should be granted access to secrets, and only when needed.
4. **Retrieval:** Apps and systems must retrieve secrets to allow access to protected services or resources in an automated, efficient and secure way.
5. **Rotation:** Secrets should be regularly rotated/ updated to reduce the risk and severity of a breach.
6. **Revocation:** When a secret is no longer needed, it should be revoked.
7. **Expiration:** Secrets should have an expiration date so they don't create ongoing vulnerabilities, and credentials should be operational for the shortest time possible.


Secrets
Gap: Unmanaged assets
Supports Vault and cloud secret managers
Vaulted
Gap: Static credentials
Dynamic
Gap: Non-rotated, out of policy
Includes secret sync
Rotated
Gap: 99% of secrets vaulted & rotated in under 30 days

system.

The 12 essential capabilities of a modern secrets management system

A modern secrets management strategy must address the security, compliance, and operational challenges of handling sensitive information in a centralized and automated way. Secrets management must do its job without impeding an organization's ability to scale and manage the growing complexity of enterprise architecture across multiple cloud platforms, applications, and services. At the same time, it can't slow down access for the wide variety of people and machines that need to use network resources. Finally, it must track everything to allow auditors and other stakeholders to see what's actually happening. Putting all that together requires 12 essential capabilities:

1. Secure secrets storage
2. Centralized management
3. Secret scanning
4. Strong encryption
5. Secrets versioning
6. Advanced access control
7. Seamless backup and recovery
8. Automated secrets rotation
9. Support for dynamic secrets
10. Integrations, APIs, and secrets sync
11. Automated key management
12. Robust auditing capabilities

Let's take a closer look at each one.

1. Secure secrets storage

A modern secrets management system must enable organizations to centrally store, access, and distribute dynamic secrets like tokens, passwords, certificates, and encryption keys across any public, private, or hybrid cloud environment. It must securely hold sensitive information using encryption and provide controls to enable authorized users' convenient access while blocking unauthorized access. The system should function amid ephemeral, widely distributed infrastructure, dynamic data-sharing requests, and zero-day cybersecurity threats.

At its core, a secrets management system should store all secrets securely in an encrypted database, helping to fight secrets sprawl. A secure centralized storage system provides consistent security optimized for complex multi-cloud environments, compared to the piecemeal protection found in traditional app-based secrets management approaches. The storage system, typically including a dashboard and command-line level administration, must enable SecOps or DevSecOps teams to securely and efficiently manage the entire secrets lifecycle, from creation to revocation.

2. Centralized management

Centralizing secrets management enables SecOps and DevSecOps teams to manage, monitor, and audit secrets from a single location, even in multi-cloud deployments with multiple keystores. A centralized approach makes management more efficient and reduces risk by allowing teams to control access more tightly. For instance, centralized secrets management avoids creating multiple workflows, so teams don't need to context switch between various applications to manage or obtain information about their secrets. It also makes it less likely that some secrets may get "lost in the cracks" between multiple systems.

A modern secrets management system should present a dashboard that empowers SecOps teams to track and safeguard secrets efficiently across the entire enterprise. Visibility across the IT estate enables secrets management best practices and policies unavailable to teams that use multiple systems or manage static secrets at the application level.

3. Secrets scanning

Organizations increasingly recognize that secrets management is essential to establishing a strong enterprise security posture. However, many enterprises still struggle with the step that comes before secrets management: finding all of the untracked secrets sprawled across their IT estates in source code, development environments, internal wikis, chat services, and ticketing systems. These unmanaged secrets represent a clear security vulnerability, as they may be exposed for long periods — even after their original purpose has become obsolete.

That's where secrets scanning comes in. A relatively new capability in secrets management, secrets scanning involves scanning code repositories and other data sources for sensitive information, such as publicly identifiable information (PII), passwords, security certificates, and access keys. Once an unmanaged secret is located, it's identified and can be immediately rotated or deprecated to maintain security.

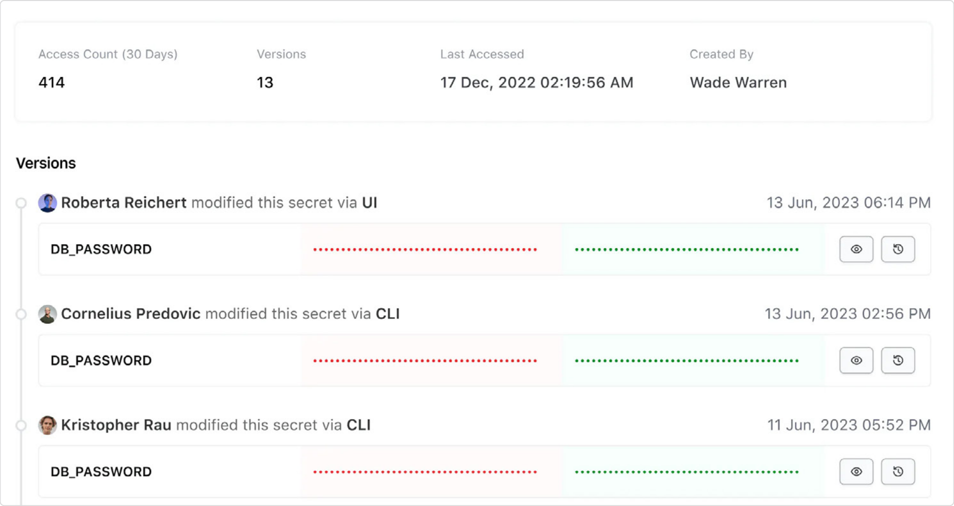
4. Strong encryption

As a general security best practice, any secrets management solution worth its salt should encrypt sensitive information, including data and secrets, both in transit and at rest. While different systems may use different encryption approaches, storing and/or transmitting unencrypted secrets in plaintext invites any person or machine to read your confidential data.

Encrypting everything is a key part of a zero trust security strategy because it prevents entities inside the firewall from reading confidential data. And it's also a key component required to meet compliance standards such as PCI, DSS, and HIPAA.

5. Secrets versioning

Seasoned SecOps teams know that managing secrets is next to impossible without the ability to "undo" changes to secrets that could affect user and machine access to networks and applications. Secrets versioning capabilities are needed to let teams roll back changes that expose security vulnerabilities or block access. Modern secrets management solutions must maintain a history of secrets and their changes, from provisioning through revocation, in order to let teams restore a previous state when necessary.



Access Count (30 Days)	Versions	Last Accessed	Created By
414	13	17 Dec, 2022 02:19:56 AM	Wade Warren

Versions

Roberta Reichert modified this secret via **UI** 13 Jun, 2023 06:14 PM

DB_PASSWORD

Cornelius Predovic modified this secret via **CLI** 13 Jun, 2023 02:56 PM

DB_PASSWORD

Kristopher Rau modified this secret via **CLI** 11 Jun, 2023 05:52 PM

DB_PASSWORD

6. Advanced access control

Now that castle-and-moat security approaches are obsolete, modern enterprises rely on identity-based access controls. Access privileges are granted to people, devices, data, or applications on a need-to-know basis. Based on zero trust and least-privilege principles, a modern secrets management system must ensure every access request is authenticated, authorized, and limited to only the specific secrets needed.

Robust access controls can also help contain the damage from a secrets leak. Secrets management systems must support and enforce the implementation of a wide variety of policy-based access controls and authentication methods, including role-based access controls (RBACs), multi-factor authentication (MFA), and fine-grained permissions. Access control policies require access privileges with defined paths utilizing tokens granted to authenticate individual users.

7. Seamless backup and recovery

Secrets are becoming increasingly ephemeral, but that doesn't mean a network outage won't wreak havoc with secrets management and user and machine access. Today's enterprises must ensure seamless, automated backup and recovery mechanisms for secrets to provide quick recovery and business continuity in the event of data loss or system failure.

To ensure availability and meet uptime requirements in high-demand industries, secrets management data should be replicated on multiple clusters for fast recovery. And the backups must be encrypted to provide security during the recovery process.

Finally, for maximum coverage, the backup and recovery capability should integrate with a broad range of enterprise systems and platforms. Secrets backup and recovery capabilities should be able to leverage common authentication protocols, such as OIDC, to authenticate and integrate with popular enterprise systems and platforms.

8. Automated secrets rotation

The longer a secret lives, the more likely it is to be leaked or compromised. To optimize security, secrets should be revoked as soon as they are no longer needed. Secrets management solutions should provide dynamic secrets on a just-in-time basis and revoke or update them after a specified time to live (TTL). This process is known as secrets rotation, and is essential to avoiding proliferation of potentially vulnerable, long-lived static secrets.

Automated and centralized secrets rotation makes the process easier to coordinate compared to a manual approach or relying on a CSP's proprietary secrets management software, which is typically focused on the providers' infrastructure and is unlikely to cover all the enterprise's secrets. Taking a cloud-provider agnostic approach helps alleviate that concern.

9. Support for dynamic secrets

The essence of a modern secrets management solution is the ability to generate dynamic secrets on demand. Dynamic secrets do not exist until needed and are revoked after a predetermined time, after they are used a specified number of times, or on demand, so there is less risk of their being compromised.

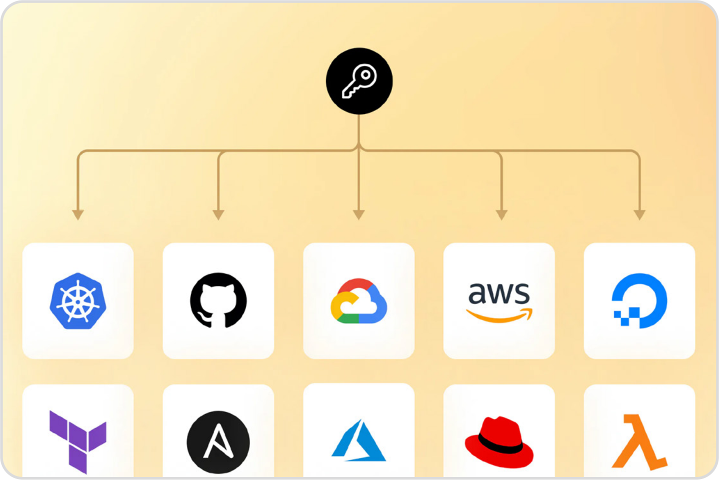
The most mature credential management practices include auto-generating credentials, such as API tokens, for each use case, allowing them to be active for only a short period. That way, the potential blast radius is greatly reduced, even if there's a leak.

Put simply, dynamic secrets:

- Minimize leaky applications by ensuring credentials are ephemeral.
- Establish unique credentials that remove the ambiguity of access, improving forensics and investigations.
- Impose automatic rotation, which improves security posture and compliance with security standards.
- Enable practical revocation, which limits the blast of a revocation, helping to prevent full-service outages.

10. Integrations, APIs, and secrets sync

One challenge to effective secrets management is that while a robust secrets management system should serve as the system of record for all secrets, not all secrets live in one place. For example, third-party applications and external APIs may hold their own secrets, which can be difficult for secrets management platforms to discover, much less manage.



A modern secrets management system should be able to integrate with secrets stores on multiple clouds and platforms (such as AWS Certificate Manager) via APIs. Open APIs and integrations with the applications and platforms used by the enterprise mean that teams can automatically and safely store and retrieve secrets no matter where they originate, without having to learn the ins and outs of every system — the secrets management platform knows the APIs so you don't have to.

Finally, the ability to sync secrets across multiple applications and secrets management systems from various cloud platforms is an operational imperative. Secret synching means that whenever a secret is updated or deleted in an application or API, that change is reflected in the primary secrets management system and vice versa. This asynchronous and event-based process should happen in seconds.

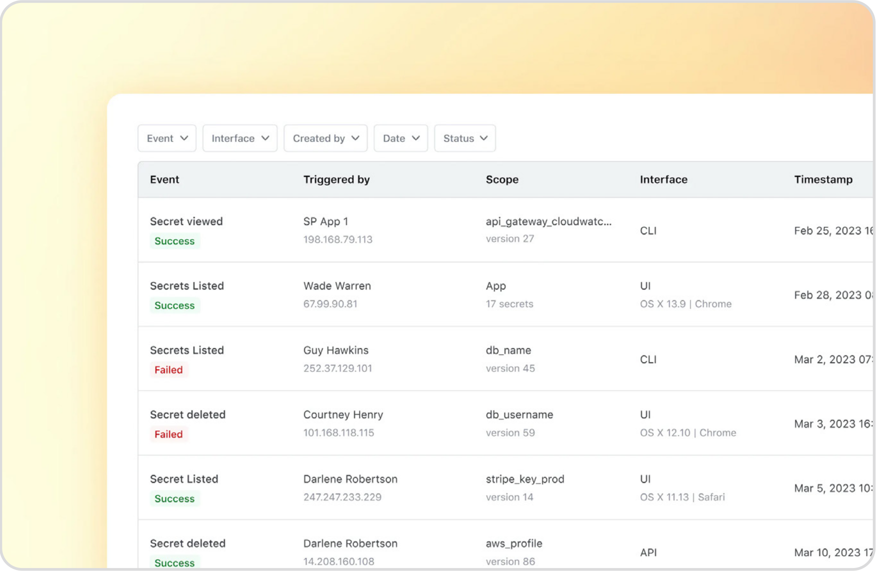
11. Automated key management

Keys are equivalent to passwords used to encrypt or decrypt data. Secure key management is required to encrypt and decrypt that data safely, and should support the use of hardware security modules (HSMs) to protect cryptographic keys. Key management is a complex issue, and automation is required to replace slow, expensive, error-prone, and hard-to-audit manual processes, especially when deployed across multiple cloud platforms.

Modern secrets management solutions should include a key management secrets engine that provides an API abstraction layer and offers a standardized workflow for the distribution and lifecycle management of cryptographic keys in various key management systems (KMS) from cloud providers, such as AWS KMS, Microsoft Azure Key Vault, and Google Cloud KMS. This allows organizations to simplify the lifecycle management of keys — even those distributed across the network — and maintain centralized control of those keys while still taking advantage of KMS providers' cryptographic capabilities. In particular, the system must be able to switch key versions on or off and rotate or delete them as needed.

12. Robust auditing capabilities

The ability to audit secrets access and administrative actions is a core element of any modern secrets management system. Secrets management solutions must include strong auditing and monitoring features to track who or what accessed secrets and when. Without robust logs, for example, SecOps teams can't trace what changed, when it was changed, or who changed it, making it challenging to identify the root cause of issues. Auditing is also required to understand whether security practices are meeting internal standards. In addition, many highly regulated industries have strict auditing requirements so that regulatory agencies can track compliance.



Event	Triggered by	Scope	Interface	Timestamp
Secret viewed Success	SP App 1 198.168.79.113	api_gateway_cloudwatc... version 27	CLI	Feb 25, 2023 16
Secrets Listed Success	Wade Warren 67.99.90.81	App 17 secrets	UI OS X 13.9 | Chrome	Feb 28, 2023 0
Secrets Listed Failed	Guy Hawkins 252.37.129.101	db_name version 45	CLI	Mar 2, 2023 07:
Secret deleted Failed	Courtney Henry 101.168.118.115	db_username version 59	UI OS X 12.10 | Chrome	Mar 3, 2023 16:
Secret Listed Success	Darlene Robertson 247.247.233.229	stripe_key_prod version 14	UI OS X 11.13 | Safari	Mar 5, 2023 10:
Secret deleted Success	Darlene Robertson 14.208.160.108	aws_profile version 86	API	Mar 10, 2023 17

Auditing requires a modern secrets management solution to keep detailed logs of all requests and record their responses, delivering the transparency needed to satisfy regulatory or compliance mandates. For auditing purposes, these logs can be analyzed with onboard information and event management (SIEM) features and/or sent to existing purpose-built log management and alerting tools.

How HashiCorp Vault delivers modern day secrets management

HashiCorp Vault is an identity-based secrets management product designed and built to meet the requirements of modern secrets management. Because most enterprises today suffer from secrets sprawl, Vault provides a central, secure place to store and manage secrets (key/value secrets, API keys, passwords, certificates, etc.) that applications need in order to work with other applications and services. Leveraging identity access management (IAM), Vault provides secure policy-based access only to authorized users or machine resources.

Vault's key features include:

Secure secrets storage: No matter which storage backend you use, Vault encrypts key/value (KV) secrets prior to writing them to persistent storage, so even if bad actors gain access to the raw storage, they still can't access your secrets.

Secrets sync: Vault can maintain a one-way sync for KV secrets into various destinations that are easier to access for some clients. Vault remains the system of record but can cache a subset of secrets on various external systems acting as trusted delivery systems.

Dynamic secrets: Vault can generate secrets on-demand for systems, such as AWS or SQL databases. For example, when an application needs to access an S3 bucket, it asks Vault for credentials, and Vault will generate an AWS keypair with valid permissions on demand. After creating these dynamic secrets, Vault will also automatically revoke them after the lease is up.

Data encryption: Vault provides encryption as a service (EaaS) to enable security teams to fortify data during transit and at rest. So even if an intrusion occurs, your data is encrypted and the attacker cannot access the raw data. Specifically, Vault can encrypt and decrypt data without storing it, which allows security teams to define encryption parameters and developers to store encrypted data in a location such as a SQL database without having to design their own encryption methods.

Leasing, renewal, and revocation: All secrets in Vault have a lease associated with them and the TTL can be specified. At the end of the lease, Vault will automatically revoke that secret. When a lease is revoked, it immediately invalidates that secret and prevents any further renewals. If desired, leases can also be renewed via built-in renew APIs.

Secrets scanning: HCP Vault Radar automates the detection and identification of unmanaged secrets so that security teams can take appropriate actions to remediate insecure secrets in the wild before they cause security issues.

Integration and extensibility: Vault is a cloud-agnostic platform with a broad range of curated official, partner, and community **plug-ins** to easily integrate with authentication methods, databases, and secrets engines. Plugins can be broken into two categories, secrets engines and auth methods. They are secure and completely separate, standalone applications that Vault executes and communicates with over remote procedure call (RPC). Vault also provides a **RESTful web API** that can be used to securely access and control every aspect of Vault.

Conclusion

In an increasingly multi-cloud world of distributed applications, services, and infrastructure, platform-agnostic secrets management is more crucial than ever. Enterprises are increasingly realizing the crucial role secrets management systems play in ensuring the safekeeping of secrets across the enterprise and throughout the software development lifecycle (SLDC). In the 2023 HashiCorp **State of Cloud Strategy Survey**, which ranked security as the top multi-cloud success factor, three out of four respondents called secrets management "important" or "very important" to the success of their cloud strategy.

But while the major public cloud providers offer proprietary secrets management solutions, these CSP-based solutions may not cover all the bases described in this paper. Most critically, perhaps, these systems naturally focus on secrets held on their platform. But in modern multi-cloud and hybrid deployments, secrets can be scattered across the entire digital estate. Managing them all could require learning, using, and maintaining multiple secrets management solutions, adding unnecessary toil and raising the risk of secrets getting lost in the cracks between systems.

Only a modern, cloud-agnostic secrets management platform — integrated and synced with the cloud providers' systems, third-party applications, and external APIs — can synchronize and secure secrets from across the enterprise in a single place while still allowing easy access to distributed workers and applications. Nothing less will do.

If you're interested in learning how **HashiCorp Vault** could support your secrets management needs, **contact our team**.

About HashiCorp

HashiCorp is a leader in multi-cloud infrastructure automation software. The HashiCorp software suite enables organizations to adopt consistent workflows and create a system of record for automating the cloud: infrastructure provisioning, security, networking, and application deployment. HashiCorp's portfolio of products includes Vagrant™, Packer™, Terraform® , Vault™, Consul® , Nomad™, Boundary™, and Waypoint™. HashiCorp offers free community source-available products, enterprise products, and managed cloud services. The company is headquartered in San Francisco, though most HashiCorp employees work remotely, strategically distributed around the globe. For more information visit HashiCorp.com.

HashiCorp

www.hashicorp.com