

QxVault™: Simplifying Secure Secrets Management

Introduction

Every organization that touches IT or cloud infrastructure is burdened with managing critical secrets, including API tokens, passwords, certificates, cryptographic keys, and more. These secrets are often scattered across developer laptops, configuration files, environment variables, source code repositories, CI/CD pipelines, and beyond. Mismanaging them can lead to unauthorized access to services and data, an outcome that can be disastrous and extremely costly for an organization.

The uncontrolled proliferation of secrets across an organization's systems is known as Secrets Sprawl. The discipline dedicated to addressing this challenge is known as Secrets Management. In broad terms, an effective secrets management solution should:

- Securely store secrets in a central location
- Securely distribute secrets to the applications, machines, and clients that need them
- Support the rotation of secrets and credentials
- Support the issuance of short-lived, just-in-time (JIT) credentials
- Enforce fine-grained, role-based access controls on individual secrets, minimizing the number of secrets any unauthorized party could potentially access
- Extend access to existing systems, such as databases, by issuing and distributing constrained, short-lived dynamic credentials to clients

A solution deployed to solve an organization's secrets sprawl problem takes on significant responsibility. A secrets management system must be highly available and able to reliably serve the applications and services that depend on it, because those applications and services rely on it to authenticate to one another and, in many cases, simply to operate. Improper deployment or operation can lead to severe consequences ranging from service outages to potential secrets exposure and data breaches. Hence, secrets management is critical infrastructure whose failure can be very costly for the organization.

Because it is critical infrastructure, a secrets management system must be deployed securely, be resilient to outages, and support zero-downtime upgrades. And while the benefits of implementing a secrets management solution are substantial, correctly setting up and maintaining one is not straightforward.

For an added layer of security, it's often recommended, and in some industries required, to incorporate a Hardware Security Module (HSM) into a secrets management deployment. Unfortunately, introducing an HSM adds significant new deployment and management complexities, because the platform engineers and IT professionals tasked with implementing this setup often have little cryptographic expertise, and even less HSM-specific expertise.

QxVault™ is a secrets management appliance with a built-in, quantum-safe, FIPS 140-3 Level 3 HSM. It helps organizations and IT professionals deploy and manage secrets instances that are secure, highly resilient, and easy to use. In this whitepaper, we explain how QxVault™™ simplifies the deployment and management of highly secure, resilient secrets management solutions.

What is QxVault™?

QxVault™ is a secret management appliance that combines a production-ready secrets management platform with a built-in, quantum-safe, FIPS 140-3 Level 3 Hardware Security Module (HSM). The platform exposes OpenBao-compatible APIs.

Normally, organizations must separately source, deploy, and integrate a secrets management platform with an HSM. QxVault™ delivers both as a single, tightly integrated appliance, with the secrets management application natively integrated with the built-in HSM. OpenBao-compatible APIs can be used to facilitate integrations with the applications, systems, and clients that need secrets.

How Does QxVault™ Simplify Secure Secrets Management?

1. Hardened, Production-Ready Solution in a Box

Quick-start deployments of proof-of-concept, software-based secrets management systems are typically straightforward and fast to stand up. These environments are well-suited for initial functional and feature exploration, but their security posture is not anywhere close to what a production system demands.

Recognizing this gap, most secrets management vendors and open-source communities publish security hardening guidelines for production deployments. These guidelines typically span a wide range of areas, including, but not limited to:

- Low-level operating system configuration
- Access management for the underlying VM or host
- Configuration of secure network connections (e.g., TLS/mTLS)
- Procedures for managing core secrets, such as root tokens and unseal keys

Turning these guidelines into a properly hardened, production-grade deployment is neither quick nor simple. In practice, the burden of implementation falls on the end-user.

QxVault™ removes this burden from the end-user by delivering hardened secrets management as a self-contained appliance, rather than as a checklist for the customer to work through. This is achieved through two complementary layers:

Hardened by default. The QxVault™ appliance ships with the operating system and secrets management platform pre-configured to meet standard hardening requirements, so the deployment starts in a production-appropriate security state from day one.

Automated best-practice enforcement. QxVault™ includes built-in automation that continuously enforces security best practices, rather than merely enabling them at install time. Examples include:

- Automatic issuance, rotation, and management of inter-cluster certificates
- Guided, policy-enforced workflows for the handling of core credentials such as root tokens

By offering "hardened secrets management in a box," QxVault™ eliminates the need for end-users to independently implement, validate, and maintain a long list of security hardening requirements. This lets teams move directly to addressing secrets sprawl across their environment, rather than spending their initial effort just getting the platform to a secure baseline.

2. API-Compatible with the Wider Secrets Management Ecosystem

A foundational requirement for any secrets management solution is the ability to integrate with an organization's existing applications and services in order to securely store and distribute their secrets. Without this integration capability, an organization may be forced to employ developers or outsource development work to bridge the gap, an approach that is often time-consuming and costly.

In practice, the ease of deploying a secrets management solution is gated as much by its ability to integrate with an existing ecosystem of applications and services as it is by the underlying security of the platform itself. No matter how advanced or secure a secrets management solution is, it delivers limited value if it cannot be readily connected to the systems that need it.

QxVault™ addresses this by offering OpenBao-compatible APIs for the management and distribution of secrets. OpenBao is a fully open-source fork of HashiCorp Vault and provides Vault-compatible APIs. HashiCorp Vault, in turn, is a widely adopted enterprise secrets management product with a large existing base of integrations across applications, systems, and clients. As a result, QxVault™ seamlessly integrates with this same broad ecosystem, out of the box and without requiring additional development effort.

Examples of supported integrations include:

- Securing connections and secrets for Kubernetes clusters
- Dynamic, short-lived credentials for popular databases (e.g., PostgreSQL, MySQL, MongoDB)
- Identity platform integrations via OIDC

QxVault™ also supports loading and running external plugins, whether maintained by the OpenBao community, HashiCorp, or third parties.

Beyond integration breadth, this compatibility carries an operational benefit: because QxVault™'s management interface and API surface mirror those of OpenBao and Vault, IT professionals who already have experience with these platforms can operate QxVault™ with little to no learning curve, eliminating one more barrier to adoption.

3. FIPS 140-3 Level 3 Quantum-Safe HSM, Already Integrated

Incorporating a certified HSM into a secrets management deployment is often recommended, and in some cases required, to meet security and compliance requirements. However, as noted earlier, integrating an HSM introduces significant complexity, particularly for IT professionals who may not have cryptographic or HSM-specific expertise. In a typical deployment, this means sourcing a separate HSM, provisioning it, and manually configuring the secrets management platform to communicate with it, all before any secrets can be secured.

QxVault™ removes this burden entirely. Every QxVault™ appliance ships with Crypto4A's built-in, quantum-safe, FIPS 140-3 Level 3 QxHSM™ that is already tightly integrated with OpenBao. There is no external HSM to source, provision, or wire up, and no integration work for IT teams to perform. From the moment QxVault™ is deployed, all cryptographic operations, including key generation, storage, and secrets encryption, are backed by certified hardware by default.

This tight integration also means QxVault™ does not rely on general-purpose configurations to connect to an HSM, a process that is often error-prone and a common source of misconfiguration in traditional deployments. Instead, the HSM is a native part of the appliance, reducing both the operational overhead and the risk of an improperly secured deployment.

4. Simplified Management of Core Secrets

Production secrets management deployments require secrets of their own. These include certificate keys, data encryption keys, and credentials; all of which must be provisioned, rotated, and managed in a safe way. Doing this correctly for a production deployment is not obvious or straightforward. Normally, the management of secrets is simplified by incorporating a secrets management solution, creating a bit of a chicken-and-egg problem when the secrets in question belong to the secrets management system itself. Compounding this, secrets management deployments are typically considered critical infrastructure that other services depend on, meaning the management of these core secrets must be handled without introducing any downtime.

QxVault™ simplifies the management of its own core secrets thanks to its tightly integrated HSM. In fact, an operator never has to touch cryptographic keys, certificates, or raw secrets in plaintext. To illustrate this, a typical secrets management cluster requires the platform admin to deploy a private PKI, and to issue, install, and manage certificates so that nodes can

authenticate and securely exchange encrypted data with one another. QxVault™ removes this burden by leveraging the cryptographic infrastructure built into the HSM, allowing operators to provision a QxVault™ cluster without ever needing to touch or manage certificates. QxVault™ also automatically rotates these certificates without ever taking nodes offline.

The same principle extends to QxVault™'s core credentials and encryption keys, which are secured by the HSM rather than left for operators to manage in plaintext.

5. No HSM Expertise Required

Integrating an HSM into an organization's secrets management system is often recommended for an enhanced security posture, and in some cases, required for compliance reasons. Setting up such an integration typically requires the operator to source, deploy, and maintain the HSM itself. The operator must also provision and manage the cryptographic material (e.g., keys) housed within it. Finally, the HSM must be securely integrated with the secrets management solution. This is a tall order, especially for integrators who are not experts in HSMs or cryptography, which is often the case for the professionals responsible for deploying and managing secrets management solutions.

QxVault™ abstracts away this complexity entirely through its tight integration with a built-in, quantum-safe HSM, eliminating the need for integrators and operators to manually source, provision, or directly manage HSMs as part of their secrets management system.

6. Simplified High Availability Cluster Deployment and Management

A secrets management deployment is often considered critical infrastructure because other services rely on it. As critical infrastructure, it is essential for such a system to be deployed in a High Availability (HA) configuration which offers the following benefits:

- **Fault tolerance:** the system remains operational, or can quickly and automatically recover, if part of it goes down
- **Horizontal performance scaling:** the system's performance can scale by distributing processing across multiple nodes, rather than relying on a single node
- **Zero-downtime upgrades:** the system can be upgraded without taking it fully offline, by upgrading nodes one subset at a time

An HA secrets management deployment involves setting up a fault-tolerant cluster: provisioning and deploying multiple nodes, often geographically distributed, that act as a single instance. Such a cluster can automatically recover if a subset of its nodes goes offline. This same property is what enables zero-downtime upgrades, since nodes can be taken offline individually, upgraded, and brought back into the cluster without the cluster itself ever going offline.

Despite these benefits, building an HA secrets management deployment is a complex task, made even more complex when an HSM is part of the deployment. For a typical secrets management solution, standing up an HA cluster can involve:

- Securely provisioning individual nodes
- Securely connecting nodes together, which typically requires managing and distributing core credentials so that nodes can authenticate to one another and encrypt data in transit
- Deploying and connecting a distributed database to back the secrets management product

When the deployment is also HSM-backed, the complexity compounds further. This typically requires first deploying a highly resilient HSM cluster of its own, and then securely integrating the secrets management cluster with that HSM cluster, a process that is especially challenging for IT professionals without HSM-specific expertise.

QxVault™ drastically simplifies the provisioning and management of HSM-backed secrets management clusters. Setting up a QxVault™ cluster does not require the operator to provision a database, stand up a separate HSM cluster, connect nodes to that HSM cluster, or manually manage and distribute credentials for nodes to securely communicate with one another. Instead, a QxVault™ cluster can be stood up in minutes, with the operator only ever needing to run a handful of commands, without touching any secrets, cryptographic material, or an HSM API.

Provisioning a single QxVault™ node requires executing just one command on the target node. Adding a node to an existing cluster requires just two commands: one run on an existing cluster node, and the other on the node being added. In both cases, this is done without the operator ever needing to touch an HSM API, or handle any secrets or credentials directly.

Conclusion

Every organization that manages or uses IT or cloud infrastructure must deal with secrets sprawl in order to securely manage the secrets that power the various components of that infrastructure. Leaked secrets caused by secrets sprawl can be extremely costly. Addressing this requires a secrets management solution that can securely store and distribute secrets to the applications and services that depend on them. But rolling out and managing a production-worthy secrets management solution, especially one backed by an HSM, is a complex and expensive undertaking.

QxVault™ removes that complexity. It lets organizations quickly and easily stand up and manage secure, HSM-backed secrets management deployments, without requiring HSM or cryptographic expertise from the IT professionals provisioning and managing it.

With QxVault™, organizations can:

- Spend far less time and effort building and maintaining secrets management infrastructure
- Get HSM-backed protection without needing in-house cryptographic or HSM expertise
- Move faster to address secrets sprawl across their IT and cloud environments

In short, QxVault™ lets teams treat secure secrets management as a solved problem, so they can get back to focusing on the infrastructure, applications, and services that actually depend on it.