Synechron

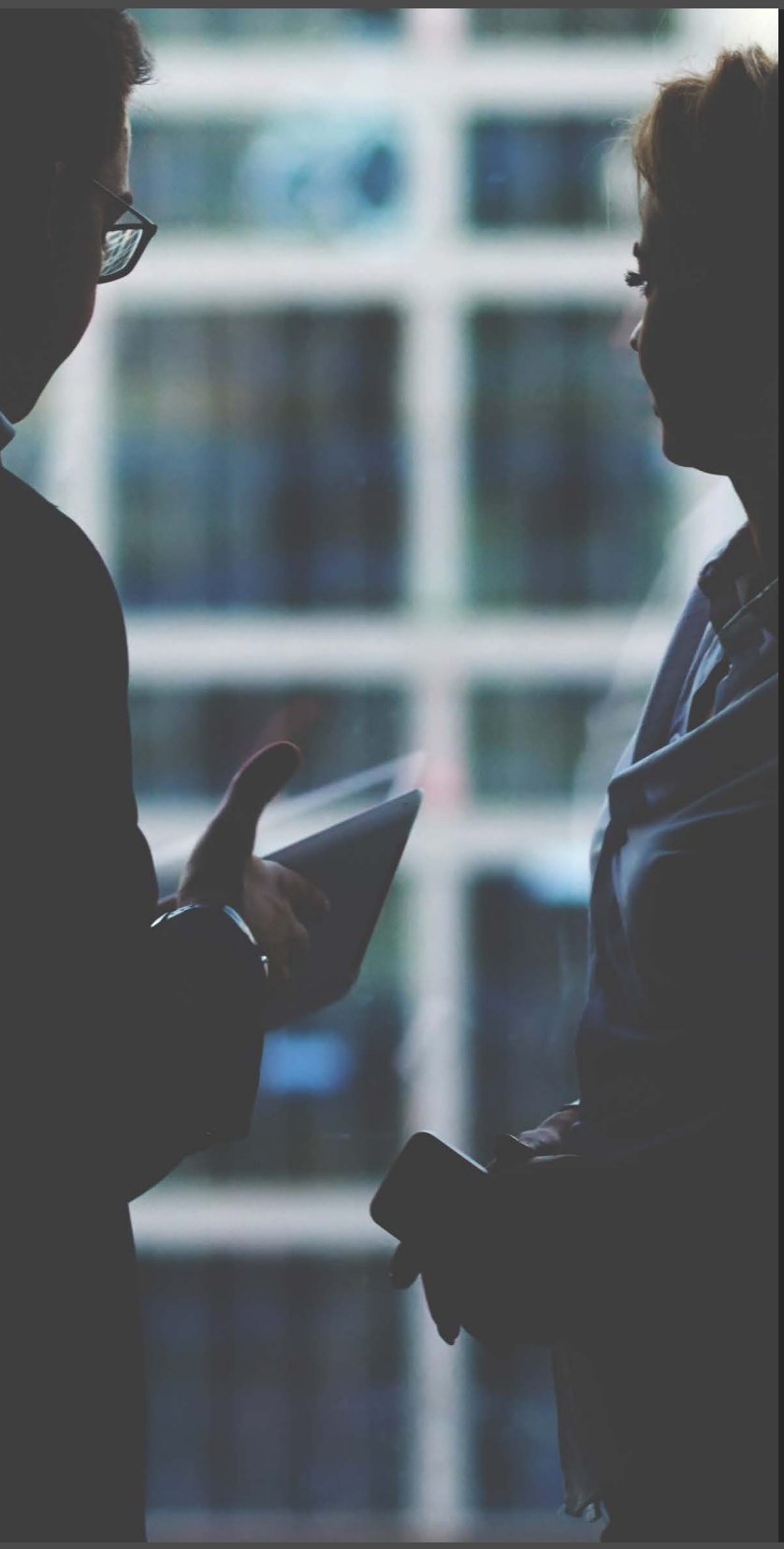
Implementing Secrets Management Using HashiCorp Vault Enterprise

Authored by:

Bobby Ip
Partner
New York

Ming Zheng
Senior Consultant
New York

Why HashiCorp Vault?



Vault is designed to solve the problem of decentralized secrets management. It is common to find secrets sprawled across multiple systems, configuration files and repositories across the organization. Often, productivity is lost in trying to build and manage secrets, integrate with Hardware Security Modules (HSMs) and other cryptographic operations.

As multi-cloud adoption becomes more prevalent, your ability to consistently secure and encrypt data across hybrid environments with HSMs will become increasingly difficult because existing tools are designed for Static Infrastructure, where you have inherent high-trust networks with clear network perimeters. In a Dynamic Infrastructure, you are working across multiple clouds and private data centers, in low trust networks, where network perimeters may span across multiple cloud providers.

Solving these use cases and filling in adoption gaps, such as with database credential management, is where HashiCorp Vault can really bring value. In general, HashiCorp Vault makes secrets management and data encryption easier, with API driven automation. Out of the box, it has extensive support for common identity providers such as Microsoft Azure, Amazon Web Services, Google Cloud, GitHub and many more. As with many HashiCorp products, your organization can also create custom authentication and secrets management engines by building customized GoLang plugins to meet any use case.

5 Things to Keep In Mind When Designing HashiCorp Vault Enterprise

#1 **Figure out your logical design first, as it will be largely dependent on where your applications, data centers, network capacity and HSM services reside.** HashiCorp Vault Enterprise is a very flexible solution, but it does have certain network performance requirements to maintain healthy states of the clusters and storage backends at load. Also, HSMs are normally very expensive and not omnipresent in large enterprise environments, which will drive how disaster recovery and BCP considerations are managed in the design. The good news is that HashiCorp Vault scales well, but there are functional differences between performance clusters and disaster recovery clusters that need to be balanced for operational manageability.

#2 **Use HashiCorp Consul as a Storage Backend. While Vault offers support for other storage options,** Consul is highly scalable and fault tolerant. It does a good job securing data at rest, while Vault secures data in transit. Underneath the hood, it uses RAFT & SERF protocols, which you'll find in products such as Kubernetes and Kafka. In future versions of HashiCorp Vault, a separate Consul specific cluster will no longer be required, which will make the installation and upkeep much easier and reduce the infrastructure footprint by at least 30%.

#3 **Use Terraform for HashiCorp Vault Policy Management.** While you can manage all Vault operations via APIs or through the Vault CLI, we highly suggest using Terraform to provision and manage your Vault Namespaces (i.e. Enablement of authentication methods and secrets engines). You can wrap governance and controls over the Terraform plans and ensure that only approved policies are deployed consistently.

#4 **Don't be scared to build custom Vault plugins, where it makes sense.** It's easy to just integrate with LDAP, but some of our clients have custom authentication and entitlement systems for various use cases that might require a custom plugin to be built.

#5 **Spend time discussing how you will manage highly sensitive secrets such as replication tokens, recovery keys and seed tokens.** Entitlements management and break glass procedures need to be carefully designed and reviewed with security and governance teams to ensure the correct control objectives are met, since HashiCorp Vault will likely contain keys to your kingdom.

Improving Adoption

HashiCorp Vault can be overwhelming with the multitude of integration options you have as an Application Owner or Platform Owner. We recommended you spend time developing common design patterns and working with your hosting platform teams to ensure that people make the right Vault adoption logic for their use case. By far, many frameworks have built libraries to support HashiCorp Vault, such as "HVAC" for Python, "Spring Cloud Vault" for Java Spring. Developers can leverage those frameworks and simply configure Vault connection strings in the property files. The frameworks will handle all the authentication for you, then you can retrieve your credentials from vault and inject into the application context when your application starts. Usually you don't need to change a lot of code to adopt vault if you are currently using Config Server to manage your app configuration, as Vault can either be used to replace Config Server or as Config Server's backend. Another popular use case for vault is for containers. Depending on your platform, you may have options that are directly supported through your orchestrator or it might be more prudent for your application to leverage a vault controller side-car pattern. Ease your team's transition by spending time developing common ways to leverage Vault.

Conclusion

We must acknowledge that large enterprises will have already deployed privileged access management tools within their organizations and have invested a huge amount of money, time and effort in wrapping governance, workflows and controls around those tools. In our client base, HashiCorp Vault is gaining an enormous amount of traction because of their API driven automation approach, their ease of integration with common CI/CD workflow tools and their ability to manage multi-cloud provider secrets. Enormous productivity gains can be realized by eliminating secrets management complexity and reducing organizational risk through minimizing secrets sprawl and bad application practices.

Authored by:

Bobby Ip
Partner
New York

To learn more about Bobby see his profile here

Reach out to:
Bobby.Ip@synechron.com

With 20+ years of IT experience delivering next-gen innovation for Tier 1 clients, Bobby works on the execution of large-scale digital and organizational transformation journeys, cloud platform adoptions, app modernization and digital workplace enablements.

Ming Zheng
Senior Consultant
New York

To learn more about Ming see his profile here

Reach out to:
Ming.Zheng@synechron.com

Ming is a senior software engineer with over 10 years of experience in IT engineering, strategy and architecture. He has broad software development expertise in building applications on public cloud. In the past, he helped a startup company build a large-scale web application which serves millions of users. He also helped a leading US bank develop the Simian Army, a tool to keep cloud environment clean, secure, resilient and compliant.

Synechron

www.synechron.com