Secrets storage

Summary

Issue

We need to store secrets, such as passwords, private keys, authentication tokens, etc.

Some of the secrets are user-oriented. For example, our developer wants to be able to use their mobile phone to look up a password to a service.

Some of the secrets are system-oriented. For example, our continuous delivery pipeline needs to be able to look up the credentials for our cloud hosting.

Decision

Bitwarden for user-oriented secrets

Vault by HashiCorp for system-oriented secrets.

Status

Decided. We are open to new alternatives as they arise.

Details

Assumptions

For this purpose, and our current state, we value user-oriented convenience, such as usable mobile apps.

- We want to ensure fast easy access on the go, such as for a developer doing on-call system reliability engineering.
- We want to be able to share some secrets among selected people, such as a team.

We are not trying to solve for single-provider, such as storing all secrets exclusively on Amazon or Azure or Google.

We do not want ad-hoc approachs such as "remember it" or "write it on a note" or "figure out your own way to store it".

Our security model for this purpose is fine with using well-respected COTS vendors, such as SaaS password management tools.

Constraints

Right now we want something that is easy i.e. no need to write code, no need to install servers, no need to make a major commitment, no need to standardize everyone.

Positions

We considered:

1. User-oriented off-the-self password managers: LastPass, 1Password, Bitwarden, Dashlane, KeePass, pass, GPG, etc.
2. System-oriented COTS password managers: AWS KMS, Vault by HashiCorp, EnvKy, Secret Server by Thycotic, Devolutions Password Server, Confidant by Lyft.
3. Sharing-oriented approaches: using a shared Google document, or shared Slack channel, or shared network folder, etc.
4. Low-tech ad-hoc approaches, such as remembering, writing a note, or relying on each user to figure out their own approach.

Argument

Bitwarden, LastPass, 1Password, and Dashlane all are commerical off-the-shelf products.

- Similar kinds of features for users, teams, organizations, etc.
- Desktop capability for Windows and Mac, and mobile capability for Android and iOS.
- Browser extensions for Chrome and Firefox, for automatic form fill in, etc.

Bitwarden has two advantages over the others:

- Bitwarden is open source, which means the security can be peer reviewed and also the company is widely-appreciated by security-oriented developers.
- Anecdotes by software workers describe a significant preference for Bitwarden over the others.

A typical good example writeup: https://jcs.org/2017/11/17/bitwarden

A typical side-by-side voting site: https://stackshare.io/stackups/bitwarden-vs-dashlane

We defer KeyPass, pass, GPG, etc. because there's additional complexity. All of these look like fine solutions for technical users. GPG looks especially good for technical users who want cross-system command-oriented capabilities.

We defer KMS because it has single-provider lock-in.

We choose Vault for system-oriented needs, because the reviews are amazingly positive, and because HashiCorp has an excellent track record fup top-quality software and support.

We veto the approaches of sharing approaches such as via shared documents, shared channels, shared network folders, etc. These do not provide the security qualities that we want.

We veto the ad-hoc low-tech approaches, because we all agree it's not a long-term path forward.

Implications

Developers may need to track secrets in two places: Bitwarden for user-oriented access, and Vault for system-oriented access.

Related

Related decisions

The decision of which CI/CD server must include proof of capability for accessing secrets.

We will need to decide how to managea the secrets, in terms of policies, rotations, organizations, etc.

Related requirements

The secrets will have related requirements for compliance, auditing, and HR onboarding/offboarding.

Related artifacts

We expect we may export some secrets to environment variables.

Related principles

Easily reversible.

Easily parallel i.e. it's easy to use a variety of password managers.

Cheap to try i.e. there's a free trial and no commitment.

Notes

Evaluation notes here. The notes are all public comments on various devops discussion boards.

Vault by HashiCorp

Vault is exactly what you want here.

Don't just throw Vault into production though, stand it up in a test environment first, because HashiCorp's documentation can be pretty lacking even if their products are amazing.

Very steep learning curve and is not trivial to stand up.

The initial setup is a bit of a pain. It's well worth it though, and the community will support it will enough for you to get by.

Horrid docs but there are lots of guides online of people setting it up and if you put a few of them together you will have a working setup.

Initial setup took fiddling with their helm charts (vault and consul). While technically you can use lots of other back-ends, I really really don't recommend it. The back-end/consul can be teeny tiny if you don't have a ton of data to store.

Definitely get comfortable/familiar with using the CLI, because the GUI is more like a proof-of-concept/advertisement portal for their enterprise edition.

The fact that you cannot just "fill it up" is a pain. For example if you have 5 fields you need to manually add each field for each item. So it's not like you pre-define fields for a specific category, and fill those fields for all

the items in that category, it's more like "you generate everything every time", which (in my mind) is a pain in the ass.

You may want to also look at goldfish as a UI for on top of vault. Makes it rather nice to get your team on board with it. They also have a demo. 1. Set up consul. 2. Set up vault pointing to consul. 3. Set up goldfish pointing to vault. 3. Setup some cron job to run consul snapshot for backups.

LastPass

LastPass Teams. We use it, has custom templates, ACL, nothing missing IMO.

I implemented LastPass at my org and give it a C+/B-. The biggest issue lately is a lack of reliability. In the past 90 days there have been multiple hours where vaults were forced into offline mode. This isn't ideal for my org due to having, literally, 4,000+ passwords stored across 20+ shared folders. As you can imagine with that many passwords at least a few get updated or added daily. We have a DR plan if issues last more than an hour or two: a script signs and encrypts a CSV dump of vault every night that can be imported into keepass.

LastPass has had unreported blips of degraded service: login 'works' but doesn't pull sites, random features broken in the admin panel, and not properly sharing keys for new top level shared folders. I have a specific 'key push'/backup user that is in every group. Usually logging in as that user will fix any key sharing issues but not when the service is degraded despite what the status page says...

For integration it can be easy if you have proper ACLs with a least privileged model e.g. if a user has read & write and read only on an entry or folder they only get read only permissions. Unfortunately my org's ACLs are not the best so I ended up using the JSON provisioning API and ~500 lines of python due to the dependent nature of our hundreds of ACLs not mapping well to the least privileged model. I ended up getting all ACLs a user was in and do a dependency walk of sorts.

If your ACL or group structure is already built with a least privileged structure in mind the AD/LDAP sync tool for Windows will work well.

Reach out to their sales team and they can hook you up with a longer Enterprise trial. Be sure you fully understand its limitations before pulling the trigger. We had a good number of growing pains but aside from outages or impairments on the server side it's been incredibly smooth.

Bitwarden

Bitwarden has a nice tooling around it (WebUI, CLI, Mobile, Desktop). Self-hosted and fairly easy to setup. Fairly good documentation and recommended tool by PrivacyTools.

EnvKey

https://www.envkey.com/ Is a saas. Really easy to implement, integrate and manage.

Features:

- Protect API keys and credentials.
- Keep configuration in sync everywhere.
- Smart, end-to-end encrypted configuration and secrets management.

- Prevent insecure sharing and config sprawl.
- Integrate in minutes.

Capabilities:

- Manage configuration and access levels for all your apps, environments, and teams in one place.
- Configure any development or server environment with just a single environment variable.

Pros:

- Good home page.
- Clear value prop.
- Visually excellent web app.
- Superior example data e.g. Algolia, AWS, Datadog, GitHub, Stripe, etc.
- Spoke with the founder for 30m about the company, UI, etc. Dane sounds well-informed, honest about the pros/cons, and a viable partner.
- The company is essentially a typical Y Combinator company, with 1 founder. Raised $120K in 2018-01.
- Focus is on getting to enterprise features, esp. moving from EnvKey cloud-hosting to either on-prem or BYOC.
- Potential path forward starting with EnvKey for ease of use, then later (or in parallel) adding Vault.

Confidant by Lyft

https://lyft.github.io/confidant/

Confidant is a open source secret management service that provides user-friendly storage and access to secrets in a secure way, from the developers at Lyft.

KMS Authentication: Confidant solves the authentication chicken and egg problem by using AWS KMS and IAM to allow IAM roles to generate secure authentication tokens that can be verified by Confidant. Confidant also manages KMS grants for your IAM roles, which allows the IAM roles to generate tokens that can be used for service-to-service authentication, or to pass encrypted messages between services.

At-rest encryption of versioned secrets: Confidant stores secrets in an append-only way in DynamoDB, generating a unique KMS data key for every revision of every secret, using Fernet symmetric authenticated cryptography.

A user-friendly web interface for managing secrets: Confidant provides an AngularJS web interface that allows end-users to easily manage secrets, the mappings of secrets to services and the history of changes.

Devolutions Password Server

https://server.devolutions.net/

Secure, manage, and monitor access to privileged accounts and sessions.

A comprehensive, highly-secured password vault that lets you control access to your privileged accounts, while also improving overall network visibility for sysadmins and providing a seamless experience for end users.

Features: centralized organization password vault, user-specific private vault, password manager, credential injection, Active Directory integration, role-based access control, two-factor authentication, enterprise ready, IP restrictions, management capabilities, automated password generator, mobile app access, password history, access reports, email alerts.

- supports data encryption
- supports multiple Authentication schemes including LDAP, O365, and Local users WITH support for MFA from multiple sources
- multiple repositories/vaults with fine-grained access controls for multiple teams
- modern Web UI
- private credential and connection vaults for personal creds/connections
- mobile apps for IOS/Android
- audit logs for each entry, who/what/when with an optional prompt for why they're accessing
- customizable templates (though they support hundreds of connection types natively)
- tons more features and a Windows/Mac thick client (Remote Desktop Manager) that you can sync to that greatly expands the options...one-click connections
- pricing isn't all that bad - up to 15 users is $500 a year for the password server

Secret Server by Thycotic

https://thycotic.com/products/secret-server/

On-premise version features:

- Total control over your end-to-end security systems and infrastructure
- Deploy software within your on-premise data center or your own virtual private cloud instance
- Meet legal and regulatory obligations that require all data and systems to reside on premise

Cloud version features:

- Software-as-a-service model lets you sign up and start right away
- Elastic scalability as you grow
- Controls and redundancy delivered by Azure with 99.9% uptime SLA

User feedback:

- We used to use that product. It was so easily bypassed and the rules only work for smart people. Lazy or dumb users can easily screw it up in a team area. Prices are negotiable when you talk to them.
- You can run it using SQL express and an Win 7box.
- Cheap.