SKA AFRICA
SQUARE KILOMETRE ARRAY

SECRETS MANAGEMENT

IN A CONTROL SYSTEM ENVIRONMENT USING HASHICORP VAULT

6TH CONTROL SYSTEM CYBER-SECURITY WORKSHOP (CS)2/HEP

PRESENTER: Anton Joubert

What is a secret

- Security-sensitive information
- Personally-identifiable information (PII)
- DB User/Pass, AWS IAM Credentials, SSL Keys, Encryption Keys
- Anything that would make the news

Content from https://speakerdeck.com/sethvargo/introduction-to-vault

How do we distribute secrets?

- How do applications get secrets?
- How do operators get secrets?
- How do secrets get updated?
- How do secrets get revoked?

Content from https://speakerdeck.com/sethvargo/introduction-to-vault

Applications

- Source code or config files
- Plaintext storage
- Git repos end up on many hard drives
- Only basic access control
- No auditing
- Very hard to revoke

```
secure  master  cat main.go
package main

const(
  mysqlUser = "root"
  mysqlPass = "s3(Ret
)
```

```
secure  master  cat config.son
{
  "mysql_user": "root",
  "mysql_pass": "s3(Ret"
}
```

Content from https://speakerdeck.com/sethvargo/introduction-to-vault

Operators

- Separate from application access
- Dropbox, Wiki, Google docs, could be anywhere…
- Zero visibility or control

Content from https://speakerdeck.com/sethvargo/introduction-to-vault

“Secret sprawl”

- Secret material is distributed
- Who has access?
- When were secrets used?
- What is the attack surface?
- What do we do in the event of a compromise?

Content from https://speakerdeck.com/sethvargo/introduction-to-vault

“Break glass” procedure

- Access Revocation
- Key Rolling
- Audit Trails

Content from https://speakerdeck.com/sethvargo/introduction-to-vault

State of the world

Many software projects, not just control systems

- Secret Sprawl
- Decentralized Keys
- Limited Visibility
- Poorly defined “break glass” procedures

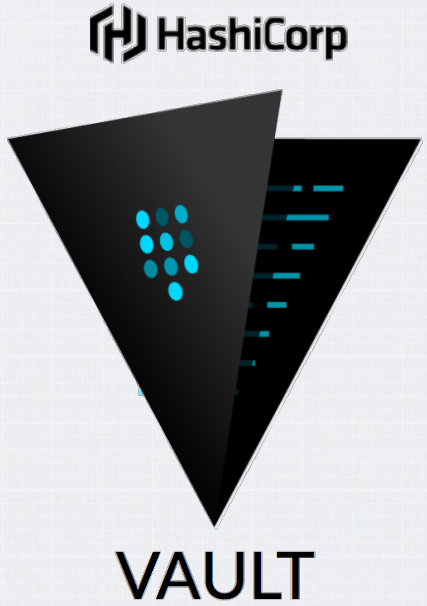
More likely in new facilities without dedicated security team and infrastructure

Content from https://speakerdeck.com/sethvargo/introduction-to-vault

Vault

"Modern" secrets management

- Single source for Secrets
- Programmatic Application Access (Automated)
- Operator Access (Manual)
- Practical Security
- Modern Data Centre Friendly (no hardware requirements)
- Free and open source (with paid option)



https://www.vaultproject.io

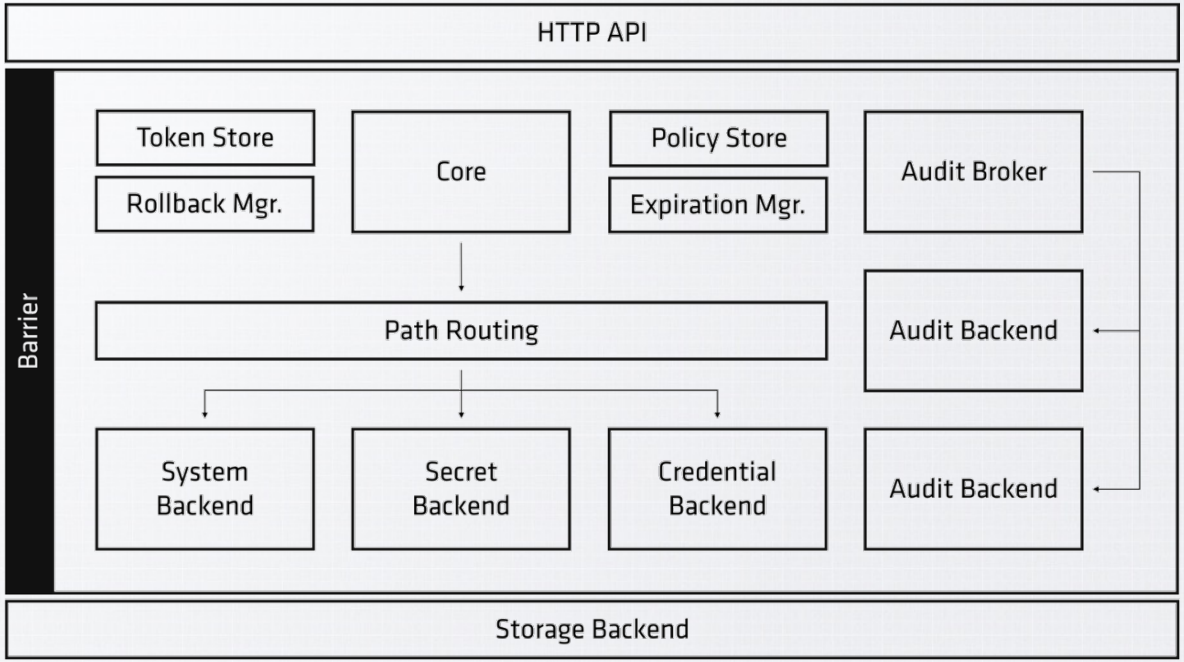
Content from https://speakerdeck.com/sethvargo/introduction-to-vault

Vault features

- Secure Secret Storage (in-memory, Consul, file, and more)
- Dynamic Secrets
- Leasing, Renewal, and Revocation
- Auditing
- Rich Access Control Lists (ACLs)
- Multiple Client Authentication Methods
- Encryption as a service

Content from https://speakerdeck.com/sethvargo/introduction-to-vault

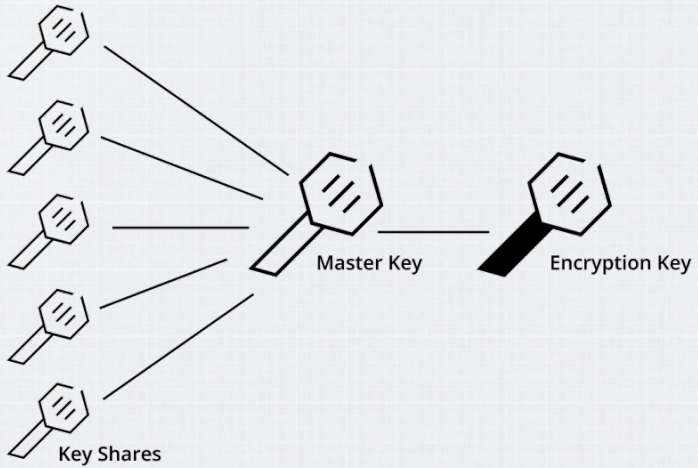
Vault architecture



Content from https://www.vaultproject.io/docs/internals/architecture.html

Vault startup

- Vault starts in “sealed” state
- Require master key to “unseal”
- Master can be shared via Shamir’s secret sharing algorithm
- An encryption key generated and stored in memory only
- Once unsealed, backends and their configurations are loaded



Content from https://www.vaultproject.io/docs/internals/architecture.html

Vault demo

(fingers crossed)

https://www.vaultproject.io/#/demo/0

Beware: using the real `vault` CLI will create a file `~/.vault-token` that you should delete when you're done, or revoke it.

Dynamic secrets

- Great for databases, cloud services like AWS, etc.
- Create users on the fly, with only limited access
- Limited time to live
- Easy to audit use
- Easy to revoke

Content from https://speakerdeck.com/sethvargo/introduction-to-vault

Access control policies

- Path based
- Wild cards (at the end only)
- Can limit on capabilities and parameter values
- Whitelist or blacklist
- IP address ranges
- Policies mapped to auth backends (user/team/application)

```
path "secret/*" {
  capabilities = ["create"]
}

path "secret/foo" {
  capabilities = ["read"]
}

path "auth/token/lookup-self" {
  capabilities = ["read"]
}
```

```
$ vault write auth/github/map/teams/default value=secret
Success! Data written to: auth/github/map/teams/default
```

Content from https://www.vaultproject.io/intro/getting-started/policies.html

So everything will be easy now, right?

Not quite…

- There is a lot to learn about Vault and secrets management in general
- Coming up with a good set of policies and how exactly to apply it to your system is hard
- Fixing old code that depended on secrets in code or config may take some work
- You will now have to trust Vault with ALL your secrets in production
- If you want to run different cluster in development and production, keeping both in sync requires effort
- When everyone knew the root password things probably seemed “easier” for them
- You need to monitor the audit logs to detect attacks and react

Enterprise version

What do you get if you pay?

- Nice web-based GUI
- Easy replication between Vault instances in multiple data centres
- Support for Hardware Security Modules
- Enterprise identities – maps different accounts from same user to single “entity”
- Multi-factor authentication

Content from https://www.vaultproject.io/docs/enterprise/index.html

Last word

Armon Dagdar (co-founder of HashiCorp) on much-publicised cyber attacks like Aurora and Stuxnet:

> *"**The key conclusion is that we cannot consider an internal network secure because of a firewall, VPN, or even air gap.** While the network perimeter is a fantastic line of defense, it shouldn't be the only one. One of the goals with Vault is to enable users to move towards a "zero trust" network, in which just being on the network does [not] imply any level of access"*

Content from https://www.infoq.com/news/2015/09/hashicorp-vault

science & technology
Department:
Science and Technology
REPUBLIC OF SOUTH AFRICA

SKA South Africa, a Business Unit of the National Research Foundation.

We are building the Square Kilometre Array radio telescope (SKA), located in South Africa and eight other African countries, with part in Australia. The SKA will be the largest radio telescope ever built and will produce science that changes our understanding of the universe

Contact information

Anton Joubert
Software Engineer – Control and Monitoring Team
Email: ajoubert@ska.ac.za