

# DevOps Secrets Management

---

```
apiVersion: scale/v18
kind: Bio
metadata:
  name: murriel
  labels:
    job: devops
    job: cloud
spec:
  containers:
    - name: orion
      image: russianblue
      command: ["cat"]
spec:
  replicas: 3
  hobbies:
    - name: making
    - name: gardening
    - name: community
```



tell me....

can your systems

keep a secret?

share a secret?

# what are secrets?

---

# personal and team secrets

- Passwords / Passphrases
- Cloud Provider Logins
- Service Provider (SaaS) Logins
- SSH Keys
- Certificates
- Kubeconfigs
- DB Credentials
- App Dashboards and Logins



User:

Password:

☐ Remember me on this computer



## Sign in as IAM user

Account ID (12 digits) or account alias

amazonaccount

IAM user name

Password

```
-----BEGIN OPENSSH PRIVATE KEY-----
```

```
b3B1bnNzaC1rZXktdjEAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABFwAAAAAdzc2gtcn
NhAAAAAwEAAQAAQEAtQ7ES36mx4/ebEqbU5sHgNEZK6LjxZod0sNcrZl+6slYAZpYewpX
lMeN6RT1s5wg247kKavlJh03cjddVfD/rjsA8L0Q8InWhIPBRg1X1Q4063TNg5vafRqGVR
pTz1DIGaXvLhmI3jCZF9HDBJiR4vhFuV1D+H4NAJx6pEUEfSa0otT0t1/atTY32cS+1E7
r4M+edKl3p2gMae/KONG3GPWKI91sRaFh84ikdL3kYa0sETKK6NLFycS57UroXIqJqcsAI
AdvoE6TZFQguQ4ASuxHiL6g+IFh914cHsREKm1QH84EvAS2KeK4E0KWr9iYnRgbs0HGup
pbgdZXW8VQAAA8hAMGRAQDBKQAAAAAdzc2gtcnNhAAABAQc1DsRLfqhbj9S5sSptTmweA0R
krqWPFmh06w1ytmX7qyVgBmlgTC1eUx43pFPWznCDBjuQpp+UmE7dyN11UV3+u0wDwvRDw
idaEg8FGDVfVDjTrdM2Dm9p9GoZVGLPPUMgZpe8uGYjeMjkX0cMeMJHi+EW5XUP4fg0AnH
qkRQR9JRsi1M63X9q1hPLfZxL6UTuvgz550qXenaAxp7806cbcy9Yo3j3WxFoWHziKR0veR
ho6wRMoro0sXJxLntRGhchAmpyWAgB2+gTpNkVCC5DgBK7EeIvqD4gWH3XhwexEQqbVafz
qS8DpYp4pTQpup2ji dGBuzQc6m1uB11dbyVAAAAAwEAAQAAQEAtMMXay/5N0qGLi ty
```

*not focusing on email passwords, computer logins, etc managed by IT*

# system\* secrets

- API Keys
- Certificates
- DB Credentials
- Encryption Keys
- Tokens
- SSH Keys
- System-to-System Authentication Secrets

## *Systems like...*



Servers  
Microservices  
Serverless functions  
Web application  
Mobile App  
On Premise App  
IoT Device Firmware  
Other machines

# why is this important?

---

information is beautiful  
World's Biggest Data Breaches & Hacks

**2019**

- Artsy
- Blank Media Games
- Capital One
- Chtrbox
- 8fit
- Australian National University
- Bulgarian National Revenue Agency
- Desjardins Group
- DataCamp
- Houzz
- Fotolog
- Ixigo
- Microsoft 44,000,000
- Petflow
- Roll20
- Suprema
- Toyota
- Whitepages
- YouNow
- WiFi Finder
- Wordguiden
- US Customs and Border Protection
- Stronghold Kingdoms
- ShareThis
- Quest Diagnostics
- Orbitz
- SKY Brasil
- T-Mobile
- Vision Direct
- WordPress
- Urban Massage
- ViewFines
- TicketFly
- Texas voter records
- Panerabread
- SingleHealth
- Quora 100,000,000
- MyHeritage
- Newegg
- MyFitnessPal 150,000,000
- Mount Olympus
- MBM Company
- High Tail Hall
- Healthcare.gov
- Grindr
- HauteLook
- EyeEm
- Ge.tt
- DearDash
- Canva
- Dubsmash 162,000,000
- BookMate
- BriansClub 26,000,000
- CoffeeMeetsBagel
- Colsonama
- CMS
- Dixons Carphone
- Click2Gov
- Dell
- Facebook
- Equifax 143,000,000
- Disqus
- CEX
- DaFont
- Bell
- Celebrity
- 2017
- Amazon
- Amazon
- Careem
- Chinese resume leak 202,000,000
- Cathay Pacific Airways
- British Airways
- Facebook 50,000,000
- LocalBlox
- GarPayNow.com
- Imgur
- Hong Kong Registration & Electoral Office
- Instagram
- Malaysian medical practitioners
- Malaysian telcos & MVNOs
- Uber 57,000,000
- Viacom
- Wonga
- Waterly
- Yahoo
- Zomato
- RootsWeb
- Snapchat
- SVR Tracking
- T10 Networks
- Swedish Transport Agency

**2018**

- Amazon
- Amazon
- Careem
- Chinese resume leak 202,000,000
- Cathay Pacific Airways
- British Airways
- Facebook 50,000,000
- LocalBlox
- GarPayNow.com
- Imgur
- Hong Kong Registration & Electoral Office
- Instagram
- Malaysian medical practitioners
- Malaysian telcos & MVNOs
- Uber 57,000,000
- Viacom
- Wonga
- Waterly
- Yahoo
- Zomato
- RootsWeb
- Snapchat
- SVR Tracking
- T10 Networks
- Swedish Transport Agency

**2017**

- Amazon
- Amazon
- Careem
- Chinese resume leak 202,000,000
- Cathay Pacific Airways
- British Airways
- Facebook 50,000,000
- LocalBlox
- GarPayNow.com
- Imgur
- Hong Kong Registration & Electoral Office
- Instagram
- Malaysian medical practitioners
- Malaysian telcos & MVNOs
- Uber 57,000,000
- Viacom
- Wonga
- Waterly
- Yahoo
- Zomato
- RootsWeb
- Snapchat
- SVR Tracking
- T10 Networks
- Swedish Transport Agency

2020 Murriel Perez-McCabe | @xmurriel 



# vectors for compromise

- Credentials in Git
- Inadvertently published secrets
  - Artifacts
  - Machine or Container Images
  - Compiled binaries
- Exposed S3 buckets
- Ex-Employees
- Internal unauthorized access
- Unauthorized password use
  - Weak passwords cracked
  - Shared/reused passwords
- Social engineering
- Network sniffing
  - Tokens/Creds sent unencrypted
- Reverse engineering

let's dream





# reality

- Decentralized infrastructure
  - Decentralized ownership
  - Hybrid Environments
  - Inconsistent process and tools
  - Path of least resistance
  - Legacy code and systems
  - Deadlines
  - Secret Sprawl
- <https://www.hashicorp.com/resources/what-is-secret-sprawl-why-is-it-harmful>
-

# anti-patterns

# hard-coded secrets

*changing history is hard*

- Secrets in git
    - Application Code
    - Config files
    - Terraform
    - Config Management Files
    - Kubernetes configs
  - Secrets in images
    - Machine images
    - Docker/container images
  - Plaintext secrets
-

# shared/re-used credentials

- Shared SSH keys
    - *“Can someone Slack me the devops-prod.pem key?”*
  - Using master keys or root users
    - Cloud accounts
    - db users
  - Creds re-used across environments
    - Dev and Prod share API keys
  - Shared kubeconfigs or client certs
  - “Default” passwords
  - No auth (mongo, elastic, redis...)
-

# insecure sharing methods

- Messaging passwords over chat
- Emailing passwords
- Passwords written publicly
- Passworded Excel sheets
- Passwords in command history
- Passwords saved where they shouldn't be

Prod DB:

root

VymEZE+Riq2Pm

Dashboard APIKey:

zaCElgL.Oimfnc8mVLWwsAawjYr4Rx-Af500Dqtlx

```
503 history
504 mysql -u root -h db.catschasingdogs.com -pSoTotallySecure
505 history
bash-3.2$
```



# ssh key problems

- Shared keys
  - Cloud provider generated keys
  - Keys embedded in images
  - Jump Box that has ALL the keys
  - Key Management
    - Managing personal keys
    - Managing system keys
    - Personal keys as machine keys
    - Global access to machine keys
    - No rotation when employees leave
-

# how can\* my systems and i keep secrets?

---

can and not *should*

# process

plus tools



~~two-factor~~

authentication

we have  
questions.

*where are your secrets?*

how to identify secrets?

where are these secrets stored?

when do we need to access them?

how do we update our secrets?

how to revoke secrets?

how is accessing these secrets?

---

# systems for secrets

# considerations

- Laws and regulations
  - Internal policies and process
  - Technical limitations
  - Access requirements
  - Resources
  - Budget
-

# useful considerations

- Human and Infrastructure Resources
  - Team bandwidth and expertise
  - Maintenance overhead
  - Existing tools
  - Buy, build, pay someone to build, or some combination
- (High) Availability and Disaster Recovery
  - Airgapped or segmented networks
  - VPN needed
  - Inter-dependent systems and points of failure
  - Centralization vs distributed
  - Footprint



# useful considerations

- AAA: Authentication, Authorization and Accounting
  - Role-Based Access Control (RBAC)
  - Access management
  - Auditing
  - Principle of least privilege
- Updates and Versioning
- Open Source Tools: Maintenance and Updates
- Backups
- Encryption: in rest and in transit
- Ease of use and ease of deployment
- Automation

# managing secrets

- Password Managers
- Secrets Storage
  - Jenkins Credentials, Rundeck Secrets, Kubernetes Secrets, Docker Secrets
  - Config Management Secrets
- Secrets Management Tools
- Cloud Provider Secrets

# password managers

- KeePass (open source)
    - KeePass, KeePassXC, ~~KeePassX~~
  - LastPass
  - 1Password
  - Dashlane
  - Keeper
  - Password Safe
  - Roboform
-

secret storage

# kubernetes

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
stringData:
  config.yaml: |-
    apiUrl: "https://my.api.com/api/v1"
    username: {{username}}
    password: {{password}}
```

- Secrets API object
- Separate secrets from configmaps, pod definitions, image
- Base 64 **encoded** not encrypted
- Encrypted in etcd
- Accessible with cluster access\*
- Role-based control
- <https://github.com/bitnami-labs/sealed-secrets>
- Helm Secrets
- Kamus

# pipeline

## Jenkins Secrets

- Credentials plugin
- <https://github.com/jenkinsci/hashicorp-vault-pipeline-plugin>

## Rundeck Secrets

## Travis CI Secrets

---

# encrypted git

encrypt secrets before committing

- git-crypt
- git-secret
- keybase
- StackExchange BlackBox
- LockGit



# github secrets

The screenshot shows the GitHub repository settings interface. At the top, a navigation bar includes links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Security, Insights, and Settings (which is highlighted with an orange underline). On the left, a sidebar menu lists various settings categories: Options, Manage access, Branches, Webhooks, Notifications, Integrations & services, Deploy keys, Secrets (highlighted with an orange bar), and Actions. The main content area is titled 'Secrets' and contains the following text: 'Secrets are environment variables that are **encrypted** and only exposed to selected actions. Anyone with **collaborator** access to this repository can use these secrets in a workflow.' and 'Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more.](#)'. Below this text, there is a section for managing secrets. It shows a single secret named 'secret\_cat' with a green lock icon and a 'Remove' button. Underneath, there is a section titled 'Add a new secret' with a 'Name' label and a text input field containing 'YOUR\_SECRET\_NAME'. A 'Value' label is also present with an empty text input field below it.

<> Code   ! Issues 0   Pull requests 0   Actions   Projects 0   Security   Insights   **Settings**

Options  
Manage access  
Branches  
Webhooks  
Notifications  
Integrations & services  
Deploy keys  
**Secrets**  
Actions

## Secrets

Secrets are environment variables that are **encrypted** and only exposed to selected actions. Anyone with **collaborator** access to this repository can use these secrets in a workflow.

Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more.](#)

secret\_cat Remove

[Add a new secret](#)

**Name**

YOUR\_SECRET\_NAME

**Value**

ALSO: <https://github.com/apps/secret-audit>



# config management

Ansible Vault

Saltstack Pillars











Chef Vault

Puppet - hiera-yaml and hiera-gpg



# secret management

# stackshare

<div><div>Vault </div><div></div><div>Secrets Management</div><div>+ Follow</div><div>Stacks 320</div><div>I Use This</div><div><div>▲ 12 Secure</div><div>▲ 8 Very easy to set up and use</div><div>▲ 8 Dynamic secret generation</div></div></div>	<div><div>AWS Secrets Manager </div><div></div><div>Secrets Management</div><div>+ Follow</div><div>Stacks 34</div><div>I Use This</div><div><div>▲ 0 Managed Service</div></div></div>	<div><div>Docker Secrets </div><div></div><div>Secrets Management</div><div>+ Follow</div><div>Stacks 29</div><div>I Use This</div><div><div>▲ 2 Secure</div><div>▲ 2 Multi-Host aware</div></div></div>	<div><div>Keywhiz </div><div></div><div>Secrets Management</div><div>+ Follow</div><div>Stacks 10</div><div>I Use This</div><div><div>▲ 1 Fuse FS</div></div></div>	<div><div>Torus CLI </div><div></div><div>Secrets Management</div><div>+ Follow</div><div>Stacks 8</div><div>I Use This</div><div>Learn more</div></div>
--	---	---	---	--

# generic secrets management pipeline

- Substitute secret with a parameter
- Secret is injected
  - at build time
  - at deploy time
  - dynamically at run-time
- Encryption at rest and in transit
  - *Depending on when they are injected they may live unencrypted somewhere*
- Determine threat model and access requirements
- Varies depending on type of application
- Sufficient encryption/obfuscation required for secrets embedded in shipped software

# open source

- Hashicorp Vault
    - Open Source
    - Enterprise
  - SecretsHub
  - Mozilla SOPS
  - Torus
  - CyberArk Conjur
  - Square Keywhiz
  - Lyft Confidant
  - Pinterest Knox
-

# Closed Source / SaaS

- BeyondTrust
    - Password Safe
    - Cloud Vault
    - DevOps Secrets Safe
  - Thycotic
    - Secret Server
    - DevOps Secrets Vault
  - CryptoMove Key Vault
-

# hashicorp vault

- robust
  - rotation and expiration
  - integrations
  - community support
  - documentation
  - professional services
- 
- management complexity
  - production requires consul and clustering and approles and policies and integrations and configuration and...
-

# Notes on Vault Implementations

- AppRoles and Policies + Rotation and Expiration considerations
- Kubernetes Vault operators
  - <https://github.com/coreos/vault-operator>
  - <https://github.com/banzaicloud/bank-vaults>
- Jenkins plugins
- API-driven
- Many integrations with config management or other systems
- [https://github.com/bruj0/vault\\_jenkins](https://github.com/bruj0/vault_jenkins)
- <https://www.hashicorp.com/resources/how-to-share-secrets-pipeline>



# cloud providers

# amazon web services

- Secrets Manager
- Key Management System (KMS)
- Amazon Certificate Manager
- Security Token Service (STS)



# google cloud

- Secret Manager
  - New! (Beta Release)
- Cloud Key Management System
- Berglas



# microsoft azure

## Azure Key Vault

- Secrets Management
- Certificate Management
- Key Management





# Let's Encrypt



<https://letsencrypt.org/>

<https://certbot.eff.org/>

# addressing anti-patterns

# *fixing\** *hard-coded* *secrets*

## Fixing Committed Secrets

- <https://securitytrails.com/blog/github-dorks>
- <https://github.com/awslabs/git-secrets>
- <https://github.com/dxa4481/truffleHog>
- <https://github.com/awslabs/git-secrets>

## Clean Up Repos

- git-filter-branch  
<https://help.github.com/en/github/authenticating-to-github/removing-sensitive-data-from-a-repository>
- BFG Repo Cleaner  
<https://rtyley.github.io/bfg-repo-cleaner/>

## Rotate Published Creds

Testing

Refactor

# *fixing shared credentials*

Implement RBAC where possible

Databases:

- tiered creds
- root, read only, read-write

Evaluate scope

- some secrets should be “shared team” secrets
- user license limitations

Refactor

---



# *fixing insecure communications*

- define securer\*  
communications channels
- enforce first-time reset if possible
- use your password manager
- encrypt and send
- secure shared location



# ssh key solutions

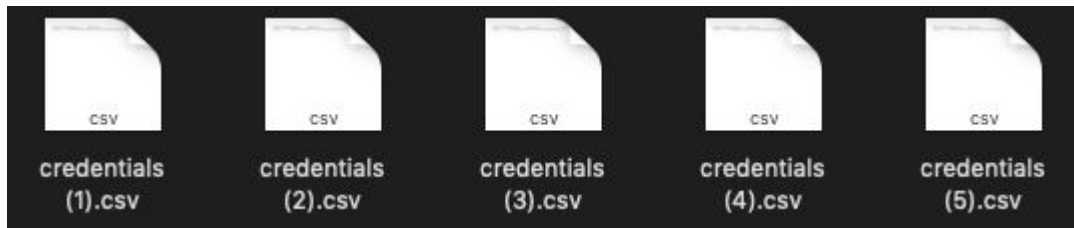
- methods for rotation
  - store / manage public keys
  - config management
  - audit images
  - audit auto-generated keys
  - use a secret manager/vault
  - remove keys when not needed
    - employee leaves
    - access no longer needed
  - limit shared keys, or access to shared keys
-

# personal and team strategies

- audit local workstation or shared systems/filesystems
- limit password re-use or password sharing
- use a password manager
- use temporary / time-limited tokens\*
- avoid root users/accounts
- **USE MULTI-FACTOR**

# personal and team policies

- awareness of secrets (visibility)
- secure sharing methods
- limit access
- password policies
  - expiration (does not necessarily work)
  - complexity
- onboarding and offboarding
  - Remove access and secrets when people leave
- password storage policies
- secret storage



# closing thoughts

- ★ knowable process
- ★ team buy-in and education
- ★ reduce barriers to usage
- ★ auditing, rotation, encryption
- ★ manage access to master keys
- ★ plaintext is bad
- ★ parameters and dynamic secrets
- ★ roles and least privilege
- ★ iterate, monitor, review

# designing secure systems

- NIST 800-63B - Digital Identity Guidelines  
<https://pages.nist.gov/800-63-3/sp800-63b.html>
- Google Cloud - User Account Management  
<https://cloud.google.com/blog/products/gcp/12-best-practices-for-user-account>
- Kubernetes Security/Secrets  
<https://github.com/kubernetes/community/blob/master/contributors/design-proposals/auth/secrets.md>
- 12-factor patterns  
<https://12factor.net/>

# resources & references

## Secrets Management Articles

- <https://www.hashicorp.com/resources/eliminating-secret-sprawl-in-the-cloud>
- <https://techbeacon.com/security/top-resources-cloud-native-secrets-management>
- <https://www.beyondtrust.com/resources/glossary/secrets-management>
- <https://dzone.com/articles/devops-and-the-proliferation-of-secrets>
- <https://techbeacon.com/security/top-resources-cloud-native-secrets-management>
- <https://blog.cryptomove.com/secrets-management-guide-approaches-open-source-tools-commercial-products-challenges-db560fd0584d>
- <https://www.hashicorp.com/resources/how-to-share-secrets-pipeline>
- <https://docs.cloudposse.com/secrets-management/anti-patterns/>
- <https://medium.com/slalom-technology/secret-management-architectures-finding-the-balance-between-security-and-complexity-9e56f2078e54>
- <https://www.praetorian.com/blog/secure-and-scalable-secret-management-in-the-cloud>

# resources & references

## Cloud Providers

- <https://cloud.google.com/blog/products/identity-security/introducing-google-clouds-secret-manager>
- <https://github.com/GoogleCloudPlatform/berglas>
- <https://aws.amazon.com/secrets-manager/>
- <https://azure.microsoft.com/en-us/services/key-vault>

## Config Management

- [https://docs.ansible.com/ansible/latest/user\\_guide/vault.html](https://docs.ansible.com/ansible/latest/user_guide/vault.html)
- <https://docs.saltstack.com/en/latest/topics/tutorials/pillar.html>
- <https://github.com/voxpupuli/hiera-eyaml>
- [https://docs.chef.io/chef\\_vault](https://docs.chef.io/chef_vault)

## Password Management

- <https://www.beyondtrust.com/blog/entry/top-15-password-management-best-practices>
- <https://hackernoon.com/we-reverse-engineered-16k-apps-heres-what-we-found-51bdf3b456bb#.io6e11q6n>



# share your secrets management

[murriel@murrielgrace.com](mailto:murriel@murrielgrace.com)

 xmurriel