Welcome to the

mongoDB®

Work from Home Webinar and Ask the Expert Series

Every Friday @ 11am PT
Full schedule to be published shortly

The Southern California team hopes you and your families are healthy and safe!

SoCal MongoDB Team

Sigfrido “Sig” Narvaez
Solutions Architect
sig@mongodb.com

John Dohoney
Solutions Architect
john.dohoney@mongodb.com

Dan Midura
Regional Director
Dan.Midura@mongodb.com

Matt Quinn
Account Executive
matthew.quinn@mongodb.com

Eric Diggins
Account Executive
eric.diggins@mongodb.com

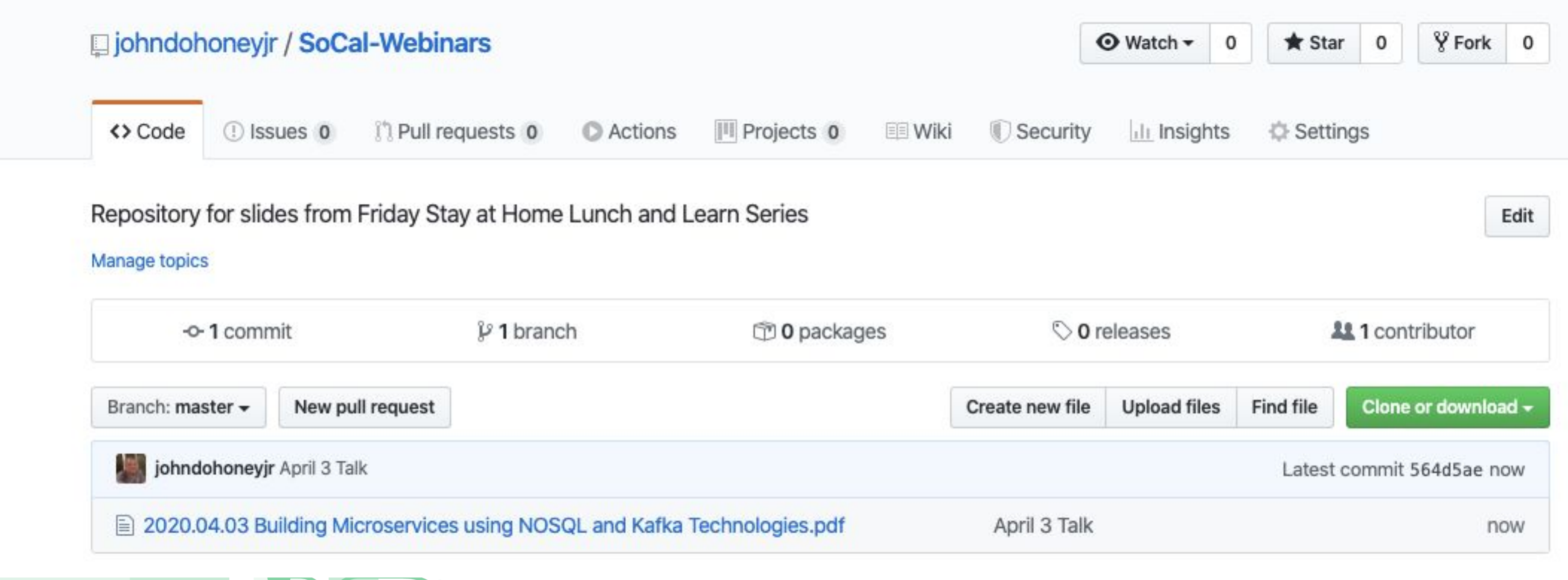
Danny Govea
Account Executive
danny.govea@mongodb.com

Miles King
Account Executive
miles.king@mongodb.com

Kyle Wilgus
Account Executive
kyle.wilgus@mongodb.com

Presentation Slides

https://github.com/johndohoneyjr/SoCal-Webinars



Repository for slides from Friday Stay at Home Lunch and Learn Series

johndohoneyjr April 3 Talk — Latest commit 564d5ae now

2020.04.03 Building Microservices using NOSQL and Kafka Technologies.pdf — April 3 Talk — now

Tech Fridays
Secure Operations with MongoDB Atlas

Using Tools for Secure and Elastic Atlas Operations

John Dohoney Jr.
Sr. Solutions Architect | MongoDB | @johndohoneyjr

Webinar Objectives

Is this a waste of time?

No, if you are using, or thinking of using MongoDB Atlas Data Platform, this webinar will help you:

1. Implement secure, elastic and self service operations for developers
2. Limit Provisioning using Atlas Role based Access, and Whitelisting to specific Projects and locations
3. Obtain TTL constrained, revocable Atlas API Keys
4. Use TTL constrained, revocable Atlas Database User Credentials
5. Securely Provision using Terraform a MongoDB Atlas Cluster
6. Use Ephemerial Database Credentials with Atlas

What tools do you need?

Hashicorp Vault — Version 1.4 or greater

This is where we will store our secrets, specifically the Atlas API keys, and Vault User tokens for Vault access

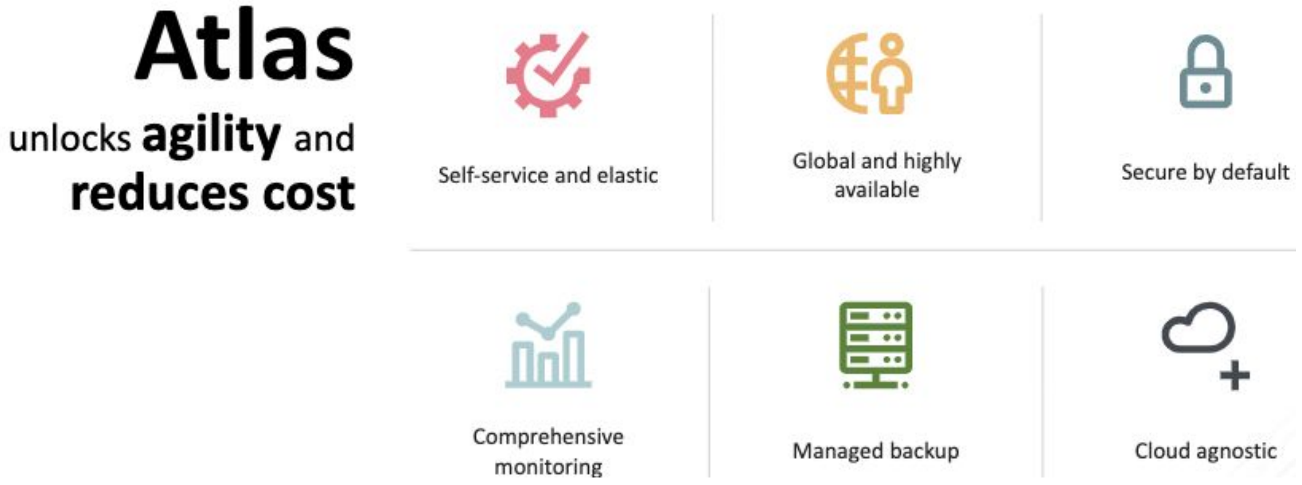
MongoDB Atlas

Your cloud data platform

Support tools

Discuss briefly other tools that might support development

What is the Atlas Cloud Data Platform



Atlas

unlocks **agility** and **reduces cost**

Self-service and elastic

Global and highly available

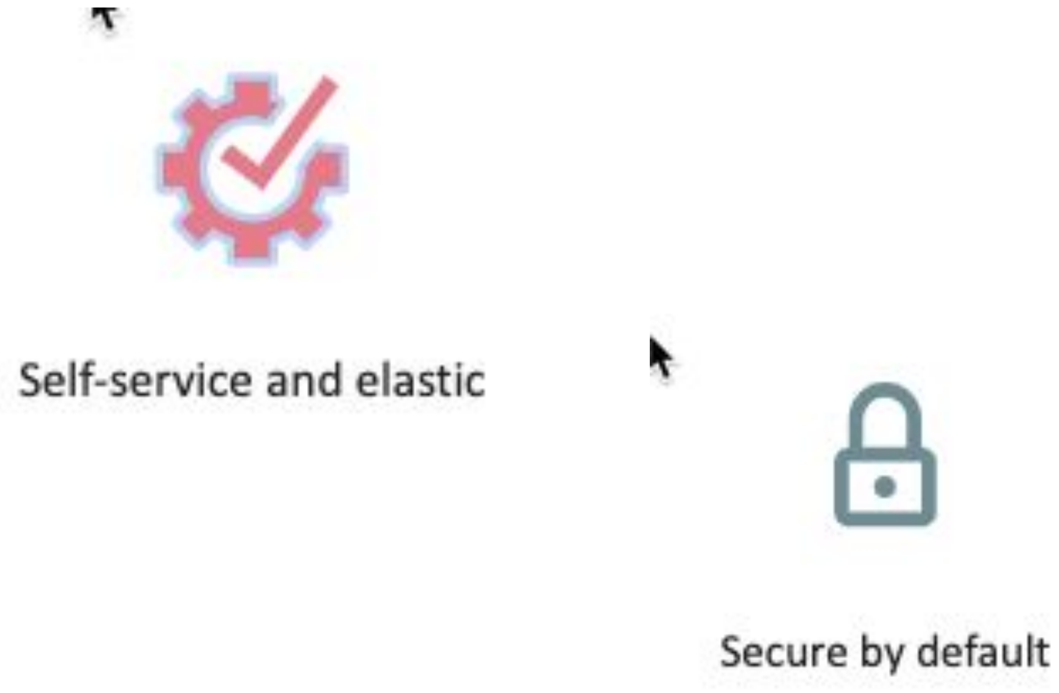
Secure by default

Comprehensive monitoring

Managed backup

Cloud agnostic

With an emphasis on ...



Self-service and elastic

Secure by default

First, some terms...

Security Terms

Vault is the **secure** place to store your companies secrets, passwords, tokens, API Keys of the system with the control of their access

Break glass (which draws its name from **breaking** the **glass** to pull a fire alarm) refers to a quick means for a person to revoke system access privileges to certain information.

TTL (Time to Live) refers to a period of time, in this case, for a credential to last.

Revocable (which draws its name from **breaking** the **glass** to pull a fire alarm) Using an identifier, the property of being able to be rescinded or removed. In this case, access to Atlas can be immediately revoked.

Vault preliminaries …

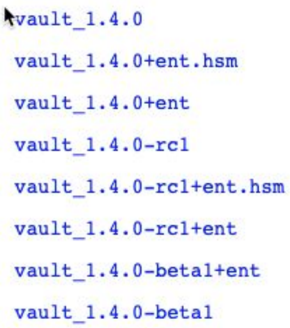
Learning Vault

https://learn.hashicorp.com/vault

Downloading Vault

https://releases.hashicorp.com/vault

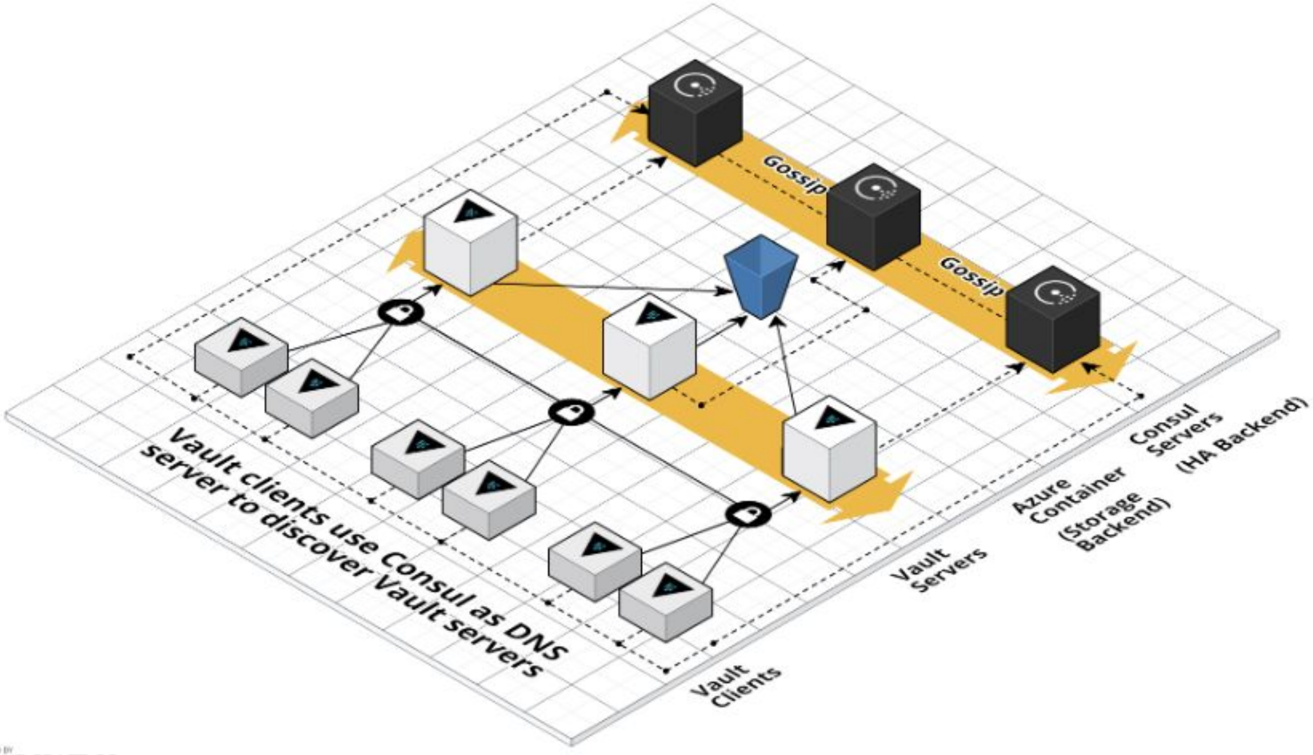
*Note, the MongoDB Atlas plugin does not exist before v1.4-beta1 (**sanity check: vault —version**)



```
vault_1.4.0
vault_1.4.0+ent.hsm
vault_1.4.0+ent
vault_1.4.0-rc1
vault_1.4.0-rc1+ent.hsm
vault_1.4.0-rc1+ent
vault_1.4.0-beta1+ent
vault_1.4.0-beta1
```

These will work, nothing before this version

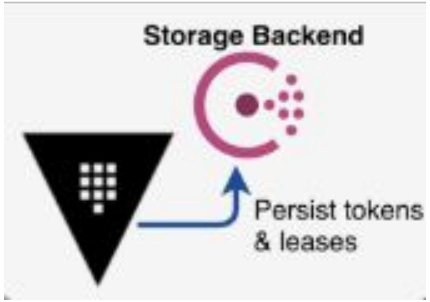
Production Vault Architecture


Gossip
Gossip
Vault clients use Consul as DNS server to discover Vault servers
Vault Clients
Vault Servers
Azure Container (Storage Backend)
Consul Servers (HA Backend)

Demo and Development Vault Architectures

Vault server -dev

Note: this is used for this demo



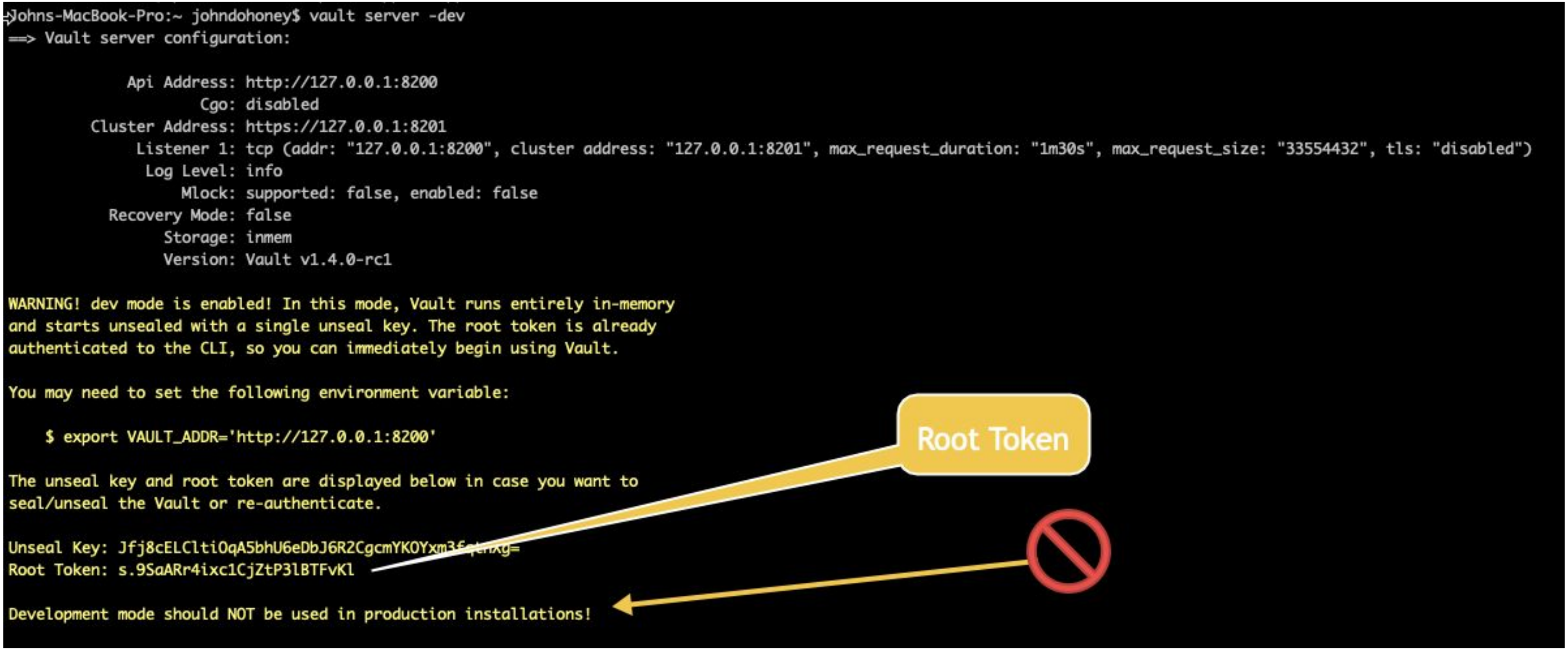
Allows for persistence to Consul, used for development or QA

Preliminary Vault Configuration

Primary Object: Get a vault token for access

Note: in Vault Server "Dev" Mode you are effectively logged in, but you still need to Add the token to all Vault Headers for API calls: X-Vault-Token

Vault Tokens — Developer Mode



```
Johns-MacBook-Pro:~ johndohoney$ vault server -dev
==> Vault server configuration:

             Api Address: http://127.0.0.1:8200
                     Cgo: disabled
         Cluster Address: https://127.0.0.1:8201
              Listener 1: tcp (addr: "127.0.0.1:8200", cluster address: "127.0.0.1:8201", max_request_duration: "1m30s", max_request_size: "33554432", tls: "disabled")
               Log Level: info
                   Mlock: supported: false, enabled: false
           Recovery Mode: false
                 Storage: inmem
                 Version: Vault v1.4.0-rc1

WARNING! dev mode is enabled! In this mode, Vault runs entirely in-memory
and starts unsealed with a single unseal key. The root token is already
authenticated to the CLI, so you can immediately begin using Vault.

You may need to set the following environment variable:

    $ export VAULT_ADDR='http://127.0.0.1:8200'

The unseal key and root token are displayed below in case you want to
seal/unseal the Vault or re-authenticate.

Unseal Key: Jfj8cELCltiOqA5bhU6eDbJ6R2CgcmYKOYxm3fqtnXg=
Root Token: s.9SaARr4ixc1CjZtP3lBTFvKl

Development mode should NOT be used in production installations!
```

Interaction with Vault

Command Line

```
vault write database/config/my-mongodbatlas-database \
   plugin_name=mongodbatlas-database-plugin \
   allowed_roles="JDTEST" \
   public_key="XKIGHEZP" \
   private_key="abeb4b3e-b4b9-4457-8f11-415713ee5ddc" \
   project_id="5d656831c56c98173cf5af4b"
```

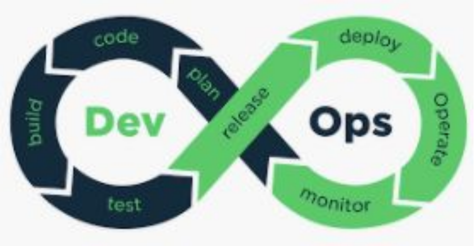
Rest API

```
curl —location —request POST
'http://127.0.0.1:8200/v1/database/config/atlas' \
—header 'X-Vault-Token: s.xKLiFh11eP6a7pVdqEa7hulp' \
—header 'Content-Type: text/plain' \
—data-raw '{
  "plugin_name"   : "mongodbatlas-database-plugin",
  "allowed_roles" : "JDTEST",
  "public_key"    : "XKIGHEZP",
  "private_key"   : "abeb4b3e-b4b9-4457-8f11-415713ee5ddc",
  "project_id"    : "5d656831c56c98173cf5af4b"
}'
```

2 Vault - Atlas Use Cases

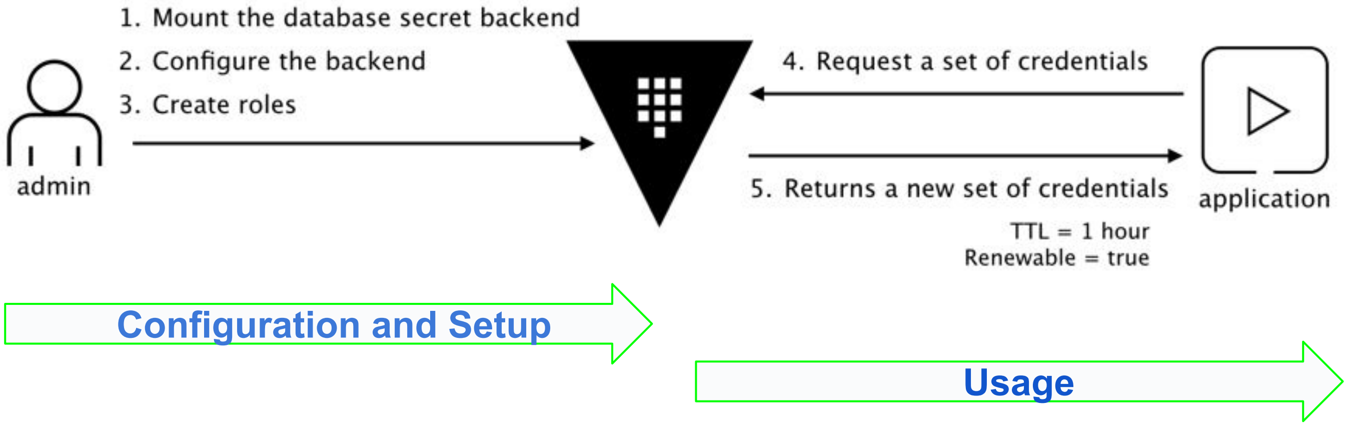
1. Dynamic Database secrets – Allows for Atlas Apps to obtain Atlas MongoDB database credentials that are:
   a. Dynamic
   b. Time Constrained
   c. Revokable

2. MongoDB Atlas Dynamic API credentials that also are:
   a. Dynamic
   b. Time Constrained
   c. Revokable



Persona’s Involved - Atlas Use Cases

1. Each Use case has a “administrator” persona that is involved in set-up
2. Although roles are different, there is a “user” or “Consumer” roles



1. Mount the database secret backend
2. Configure the backend
3. Create roles

4. Request a set of credentials

5. Returns a new set of credentials

TTL = 1 hour
Renewable = true

Terraform - API Demo

Secure Provisioning

Passwords are a problem...

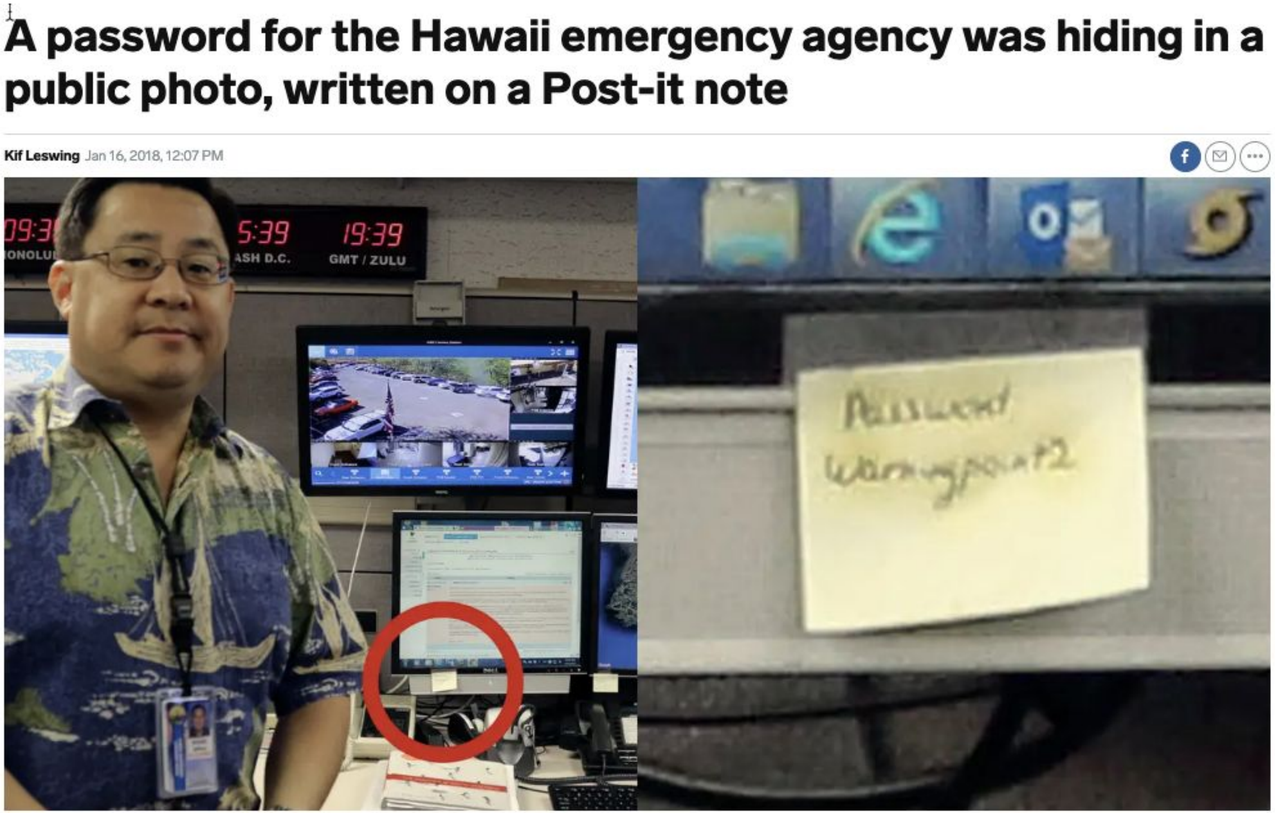
PASSWORD
123456
123456789
ROMANCE
MYSTERY
12345
SHADOW
102030
TIGGER
SUNSHINE
BOOKWORM
DRAGON
READING
AJCUIVD289
READER
PURPLE
MAGGIE
BOOKLOVER
PEANUT
CHARLIE
PRINCESS
123
MICHAEL
PEPPER
TAYLOR
1234
ANGELS
UNICORN
BUSTER
VAMPIRE
MATTHEW
CHOCOLATE
123123
SAMANTHA
12345678
FLOWER
BANDIT
654321
242424
121212
ANDREW
SOPHIE
MOTHER
ABC123
JOSHUA
MIDNIGHT
BAILEY
MORGAN
POOHBEAR
COOKIE
LIBRARY
BUTTERFLY
GINGER
1234567
CALLIE
SUMMER
TEDDYBEAR
JENNIFER
JASPER
HANNAH
WRITERSPACE
JUNIOR
WRITER
HUNTER
AMANDA
PEACHES
WINNER
FREEDOM
DANIEL
101010
PATRICK
CASPER
MURPHY
JESSICA
GEORGE
PANDORA
CRICKET

Then we add password policies, and increase the number of systems we need to access.

This is one result ...

It gets worse...

- A false alert warning of an inbound missile was broadcast in Hawaii on Saturday.
- Since then, people have discovered that a photo taken in Hawaii's Emergency Management Agency for a news article in July includes a sticky note with a password.
- Hawaii says the alert was sent was because "an employee pushed the wrong button," not because of a hack, but the photo has sparked criticism about the agency's level of security.

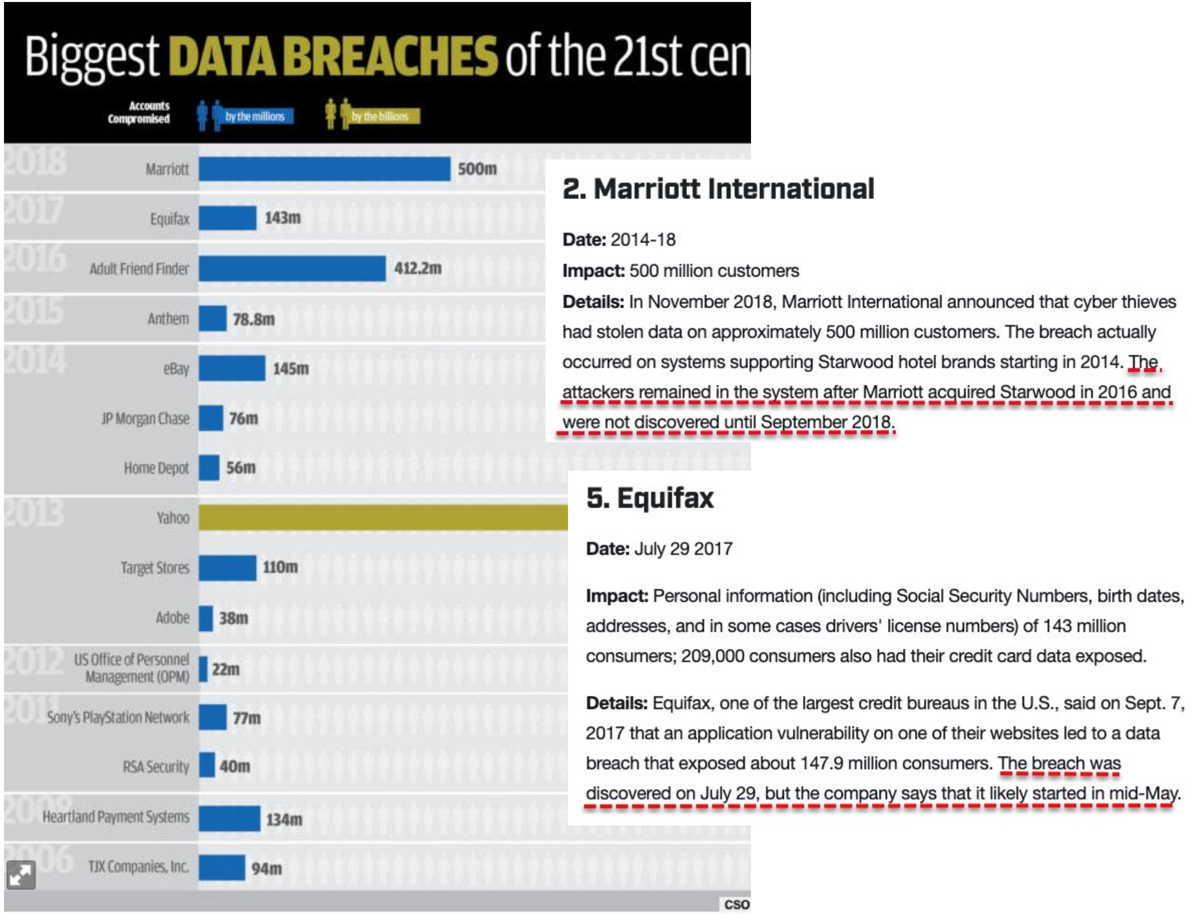


A password for the Hawaii emergency agency was hiding in a public photo, written on a Post-it note

Kif Leswing Jan 16, 2018, 12:07 PM

Illegal Systems access happens ...

Even the best companies are vulnerable



2. Marriott International

Date: 2014-18

Impact: 500 million customers

Details: In November 2018, Marriott International announced that cyber thieves had stolen data on approximately 500 million customers. The breach actually occurred on systems supporting Starwood hotel brands starting in 2014. The attackers remained in the system after Marriott acquired Starwood in 2016 and were not discovered until September 2018.

5. Equifax

Date: July 29 2017

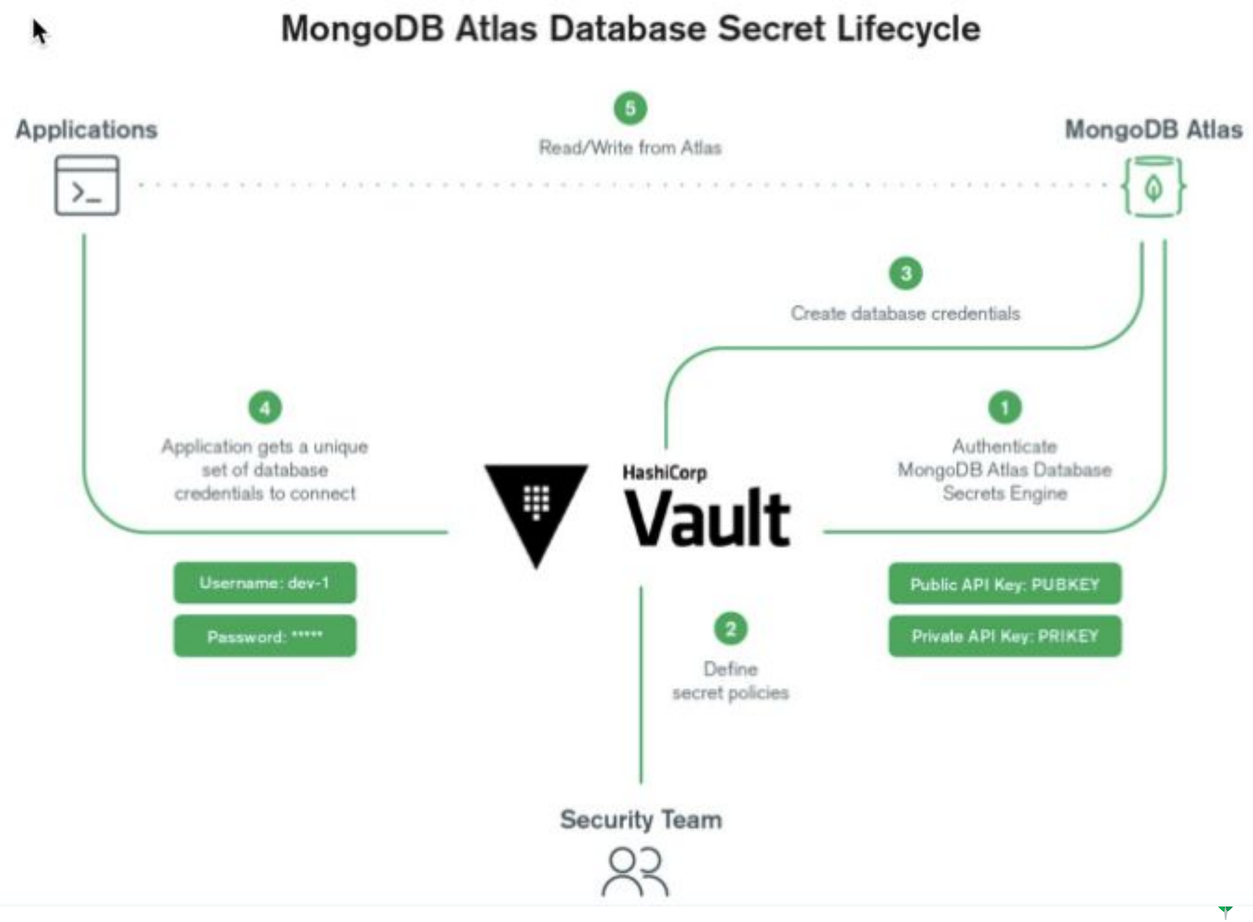
Impact: Personal information (including Social Security Numbers, birth dates, addresses, and in some cases drivers' license numbers) of 143 million consumers; 209,000 consumers also had their credit card data exposed.

Details: Equifax, one of the largest credit bureaus in the U.S., said on Sept. 7, 2017 that an application vulnerability on one of their websites led to a data breach that exposed about 147.9 million consumers. The breach was discovered on July 29, but the company says that it likely started in mid-May.

How do Dynamic Secrets Work?

Vault grants tokens for access, not credentials, to ensure least privilege with security and ensure break glass functionality

- Set expiry to ensure that access is automatically revoked on time
- Revoke outstanding access at will
- Break glass functionality to halt all access in critical situations


MongoDB Atlas Database Secret Lifecycle
Applications
5
Read/Write from Atlas
MongoDB Atlas
3
Create database credentials
4
Application gets a unique set of database credentials to connect
HashiCorp Vault
1
Authenticate MongoDB Atlas Database Secrets Engine
Username: dev-1
Password: *****
Public API Key: PUBKEY
Private API Key: PRIKEY
2
Define secret policies
Security Team

Dynamic Secrets
Big Ideas

Privilege Separation – Separation of privilege refers to the compartmentalization of privileges across various application or system sub-components, tasks, and processes.

Vault Implementation -- Create a specific policy based on role that in this case, compartmentalizes database access

Privilege Bracketing – Elevate privileges on an as-needed basis for specific applications and tasks only for the moment of time they are needed, without requiring administrative credentials or exposing passwords.

Vault Implementation -- Dynamic Secrets that utilize a Time-based “Lease” for access

Non-repudiation – You are who you say you are and it can be audited

Vault Implementation -- Vault provides metadata and audit logging for traceability

Atlas Database Secrets Demo

mongoDB®

Thank you

That’s all folks