Secrets at Scale

Automated Bootstrapping of Secrets and Identity in the Cloud

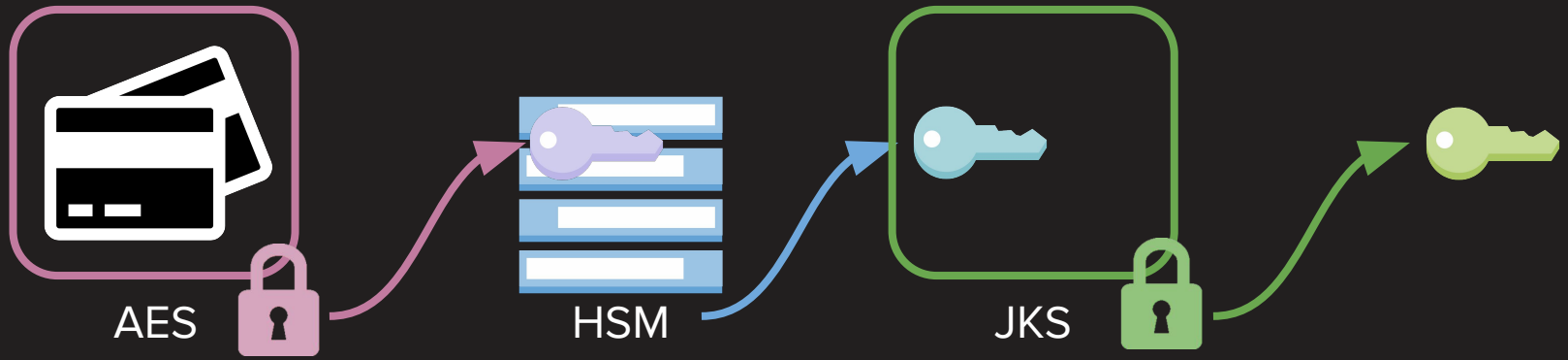


ENIGMA

Ian Haken
@ianhaken
January 30, 2017

NETFLIX

The Problem With Secrets



“Where do I put my secret?”

Secrets at Scale

- TLS/HTTPS Certificate Private Keys
- RDS passwords
- HMAC keys
- Encryption keys for credit card data, personally identifiable information, etc.
- Third-party API credentials
- Basically, anything your application needs to startup or be functional.



Secrets at Scale

Services at Netflix are

- Autoscaling
- Ephemeral
- Self-healing



Naive Solutions

- Manually copy a secret/config file after the instance is booted?
  - No way to scale!
- Just encrypt the secrets?
  - How do instances get the decryption key?
- Host the secret somewhere at a hidden URL?
  - Now that hidden URL is a secret that needs to be protected...

Most solutions just change what secret you're protecting. And if you're protect one secret with another secret...

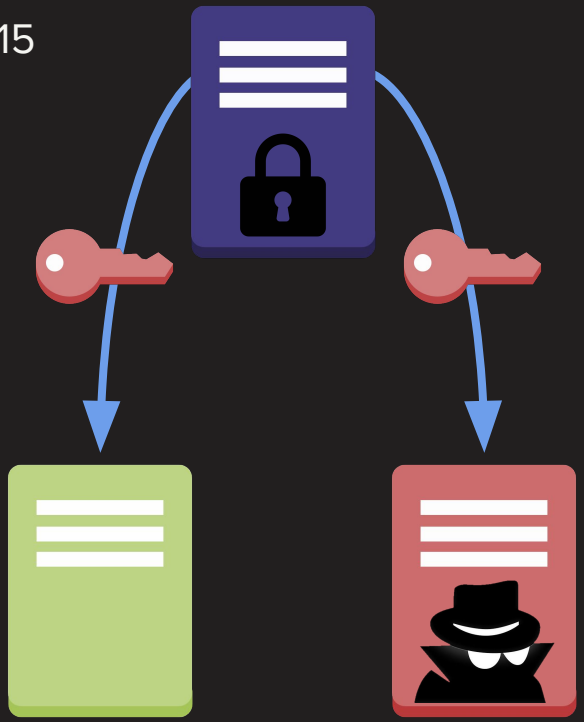
It's turtles all the way down...

Turtles All the Way Down:

Storing Secrets in the Cloud and the Data Center

Daniel Somerfield | ThoughtWorks, AppSec USA 2015

- Encrypted secrets in source
  - Blackbox, GitCrypt, Transcrypt
- Secrets managed by orchestration tools
  - Chef Vault, Ansible Vault
- Secrets fetched from a Secret Service
  - Hashicorp Vault, Square Keywhiz



> “Before performing any operation with Vault, the connecting client must be *authenticated*.
> … it is important to understand that authentication works by verifying your identity and then generating a token to associate with that identity.”

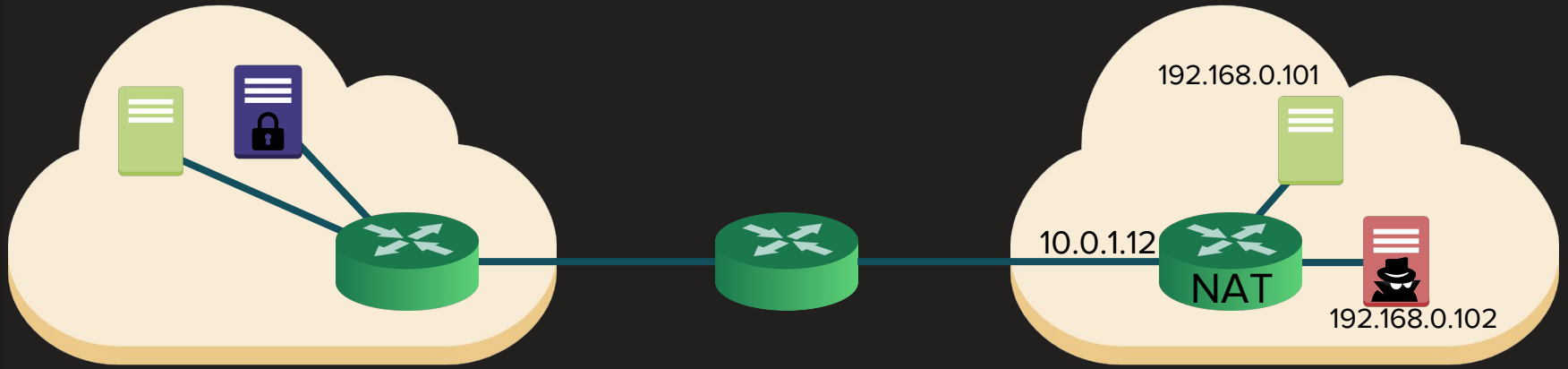
The Identity Problem

- Traditional remote authentication schemes:
  - Username and password
  - Client Token / Secret
  - HMAC with an authentication token
  - TLS Certificate and Private Key
- All these schemes involve proving possession of a secret...
  - ...making this turtle *n+1*.



Solving the secret storage problem means we need to solve the bootstrap identity problem.

Why Not IP For Identity?



VLAN hopping, ARP poisoning and Man-In-The-Middle Attacks in Virtualized Environments

Ronny L. Bull, Jeanna N. Matthews, Kaitlin A. Trumbull

https://media.defcon.org/DEF%20CON%2024/DEF%20CON%2024%20presentations/DEFCON-24-Bull-Matthews-Trumbull-VLAN-Hopping-ARP-MITM-in-Virtualized-WP-UPDATED.pdf

Remote Attestation

- In the cloud, our provider knows what application images are running where. This means the cloud provider can facilitate *remote attestation*.
  - In AWS, instances can request a metadata document signed by AWS.
  - This document is unique to each EC2 instance that calls it and can we used to prove what code (AMI) is running.



Who Are You?




```
http://169.254.169.254/latest/
  dynamic/instance-identity

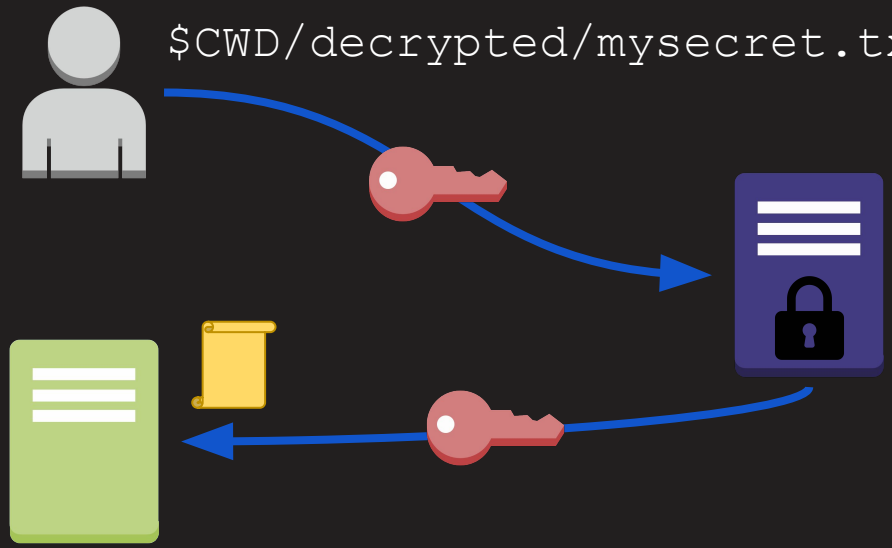
{
  "document" : {
    "privateIp"  : "10.16.112.84",
    "region"     : "us-east-1",
    "instanceId" : "i-1234567890",
    "accountId"  : "123456789012",
    "imageId"    : "ami-5fb8c835",
    "kernelId"   : "aki-919dcaf8",
  },
  "signature" : "lyoYVBoUYrY9n..."
}
```

```
{
  "securityGroups" : {
    ...
  },
  "iamRole" : "test::creditCardSrv"
  "user-data" : {
    "appName" : "creditCardService",
    ...
  }
}
```

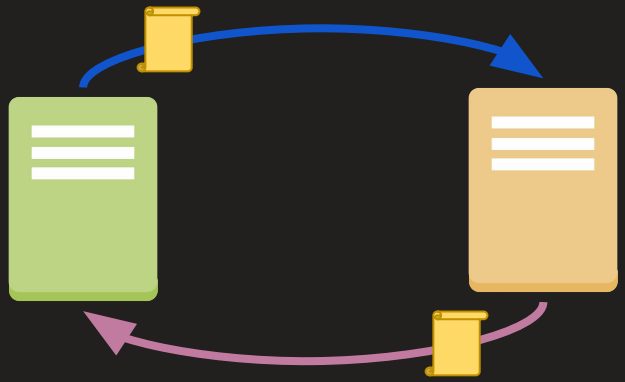
The cloud provider supplies a signed document which provides a cryptographic assertion of instance identity.

Additional metadata APIs let use map this to an internal application name and other features.

The Developer Experience



Universal Identity



```
Certificate:
  Data:
    Issuer: CN = Secret Service CA
    Subject: CN = creditCardService
    ...
```

```
Certificate:
  Data:
    Issuer: CN = Secret Service CA
    Subject: CN = userBillingService
    ...
```

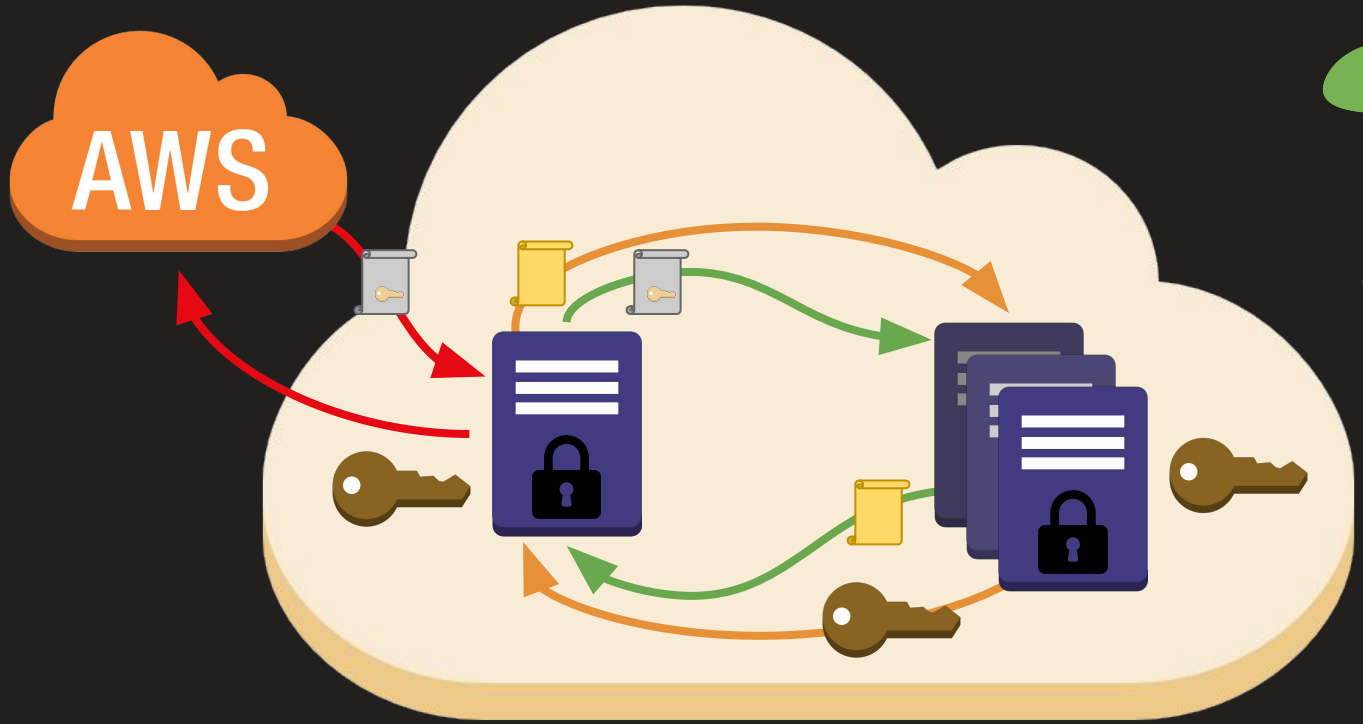
The Last Turtle

- With these tools, we've accomplished our goals:
  - Applications can get their secrets automatically
  - Only applications ever see their secrets
- Except... how does the secret server come up?





The Last Turtle





Summary

- Solving the secret storage problem meant that we had to solve the problem of bootstrapping identity as applications start up.
  - But as a bonus, this identity is re-usable throughout the ecosystem.
- The Secret Service itself is also a Secret Service client and uses it to bootstrap its own master key.
  - This makes the end-to-end solution auto-scalable and self-healing!
- We now have a clear, simple answer to the question “Where do I put my secret?”
  - Put it in the secret service…
  - …and it will automatically show up on your application’s disk.

Questions?



Twitter: @ianhaken
ihaken@netflix.com