Managing Secrets

Tom Maher
30 May 2018

Problem

Has this ever happened to you?

```
89  > library(stringr)
90  Warning message:
91  package 'stringr' was built under R version 3.1.3
92  >
93  > timestart <- Sys.time()
94  > # connect to jet-data on sql-ops
95  > j2x <- odbcConnect(dsn =
96  +                    uid =
97  +              pwd =
98  >
99  > last_monday_midnight <- floor_date(Sys.Date(), 'week') - 6
100 > last_midnight_formatted <- paste(last_monday_midnight %>% month,
101 +                          last_monday_midnight %>% day,
102 +                          last_monday_midnight %>% year,
103 +                          sep = '-') %>%
104 +     paste0("'", ., "'")
```

```
9     "userName": "$username",
10    "password":
11    "confirmPassword":
12    "zipCode": "
13  }
14  "@
15  $response = Invoke-WebRequest -Method Post -Uri "$baseUrl/api/account/register -Body $body
16  {1}
17  $body = password="                          client_id=$clientId;client_secret=$clientSecret;
18  $response = Invoke-WebRequest -Method Post -Uri "$baseUrl/Token -Body $body
19  $access_token = (ConvertFrom-Json $response.Content).access_token
20  {1}
21  $response = Invoke-WebRequest -Method Get -Uri "$baseUrl/api/members/member-info{1}
22  $clientTicket = (ConvertFrom-Json $response.Content).clientTicket
23  {1}
24  $body = @"
```

```
13  dead_letter_queue.enable: false
14  http.host: "127.0.0.1"
15  http.port: 9800-9900
16  # fatal, error, warn, info, debug, trace
17  log.level: debug
18  log.format: plain
19  # log.format: json
20  path.logs: /var/log/logstash
21  # path.plugins: []
22
23  xpack.monitoring.elasticsearch.username: logstash_system
24  xpack.monitoring.elasticsearch.password: "                    "
25
```

Overview

Problem
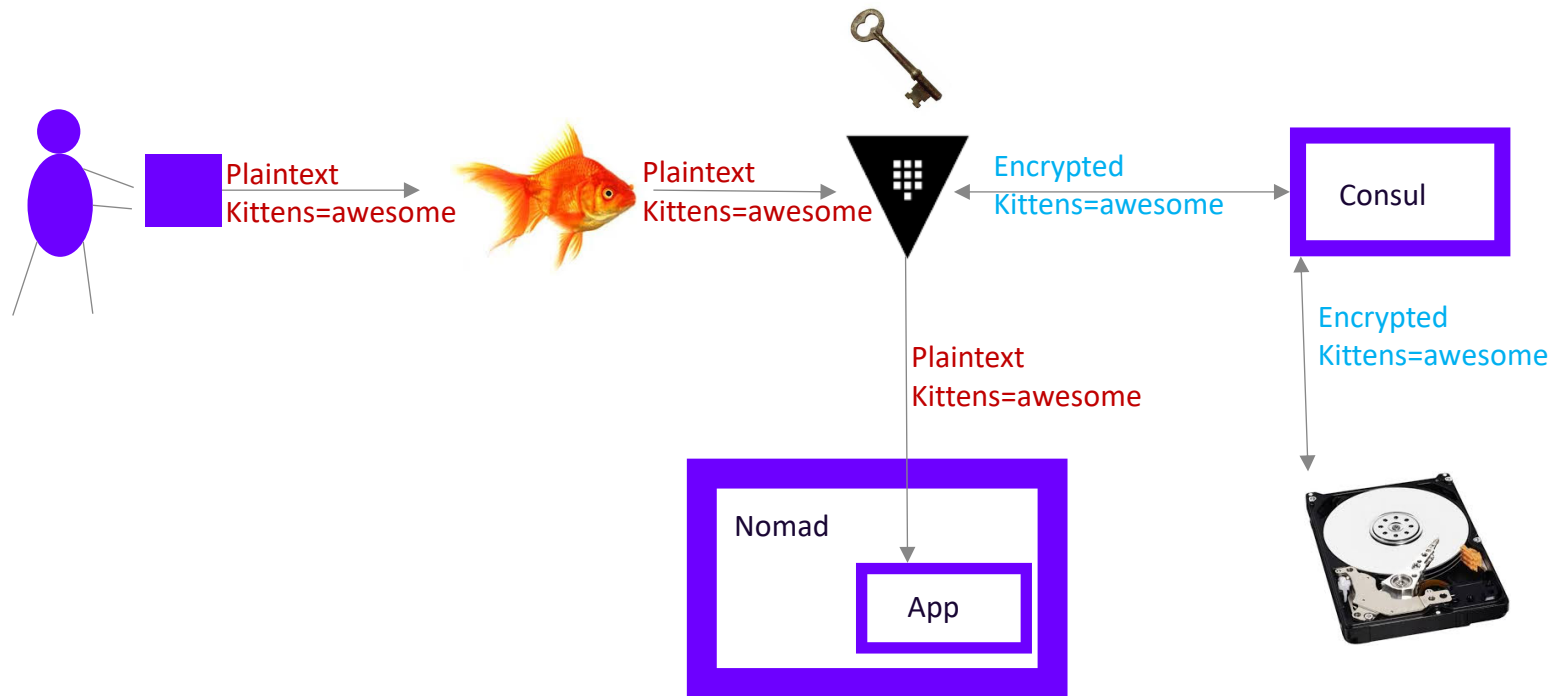
Storage

Demo – Deploy an App

Threat Model

Related Work

Migration

The Future! / Questions?

Storage

Plaintext
Kittens=awesome

Plaintext
Kittens=awesome

Encrypted
Kittens=awesome

Consul

Plaintext
Kittens=awesome

Encrypted
Kittens=awesome

Nomad

App

Demo
Deploy an application

Threat Model

- Source code leaks

- File / Directory Traversal (in app)

- Backing store leak

- Untrusted Network (Passive and MitM)

- RCE (different account, non-privileged)

- Unknown/undocumented credential roll process

Threat Model, NOT

- RCE, same account or privileged

- Malicious/compromised engineers, Dev &| Ops

- Malicious/compromised hosting provider, e.g. Azure, AWS

- Leaked environment variables

Apache Environment

Variable	Value
db_host	nas-ho.me
HTTP_HOST	localhost
HTTP_CONNECTION	keep-alive
HTTP_CACHE_CONTROL	max-age=0
HTTP_UPGRADE_INSECURE_REQUESTS	1
HTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.84 Safari/537.36
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
HTTP_ACCEPT_ENCODING	gzip, deflate, sdch
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.8,it;q=0.6
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
SERVER_SIGNATURE	no value
SERVER_SOFTWARE	Apache/2.4.23 (Unix) PHP/7.0.9

Related Work

- Nomad environment variables

- Consul and other plaintext databases

- Keywhiz

- Cloud provider

  - Azure Key Vault

  - Amazon KMS and Secrets Manager

Migration

- Detect: Scanner finds candidate secret-in-code

- Verify

  - Auto-verifier – can we authenticate somewhere with this?

  - Human – are we sure this isn't a unit test?

- File Issues with Secret Cannon

- Team playbooks how to rotate the secret

- Secret moves into Vault

- Secret Rotates

- ☺

The Future! / Questions?

Nomad Worker

Plaintext
Kittens=awesome

Application
uid=456

TCP/HTTP

Unix Domain
socket

Vault Client
uid=123

- No environment variables
- Domain sockets are immune to directory traversal
- Inspired by git-credential-cache