Managing Secrets

Tom Maher

01 May 2018

Problem

Has this ever happened to you?

```
> library(stringr)
Warning message:
package 'stringr' was built under R version 3.1.3
>
> timestart <- Sys.time()
> # connect to jet-data on sql-ops
> j2x <- odbcConnect(dsn =
+                    uid =
+                    pwd =
>
> last_monday_midnight <- floor_date(Sys.Date(), 'week') - 6
> last_midnight_formatted <- paste(last_monday_midnight %>% month,
+                                  last_monday_midnight %>% day,
+                                  last_monday_midnight %>% year,
+                                  sep = '-') %>%
+   paste0("'", ., "'")
```

```
  "userName": "$username",
  "password":
  "confirmPassword":
  "zipCode": "
}
"@
$response = Invoke-WebRequest -Method Post -Uri "$baseUrl/api/account/register -Body $body
{1}
$body = password="                                client_id=$clientId;client_secret=$clientSecret}
$response = Invoke-WebRequest -Method Post -Uri "$baseUrl/Token -Body $body
$access_token = (ConvertFrom-Json $response.Content).access_token
{1}
$response = Invoke-WebRequest -Method Get -Uri "$baseUrl/api/members/member-info{1}
$clientTicket = (ConvertFrom-Json $response.Content).clientTicket
{1}
$body = @"
```

```
dead_letter_queue.enable: false
http.host: "127.0.0.1"
http.port: 9800-9900
# fatal, error, warn, info, debug, trace
log.level: debug
log.format: plain
# log.format: json
path.logs: /var/log/logstash
# path.plugins: []

xpack.monitoring.elasticsearch.username: logstash_system
xpack.monitoring.elasticsearch.password: "                    "
```

Overview

Problem

Threat Model

Related Work

Storage

Demo – Deploy an App

Migration

Demo – Cannon Fodder

The Future! / Questions?

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

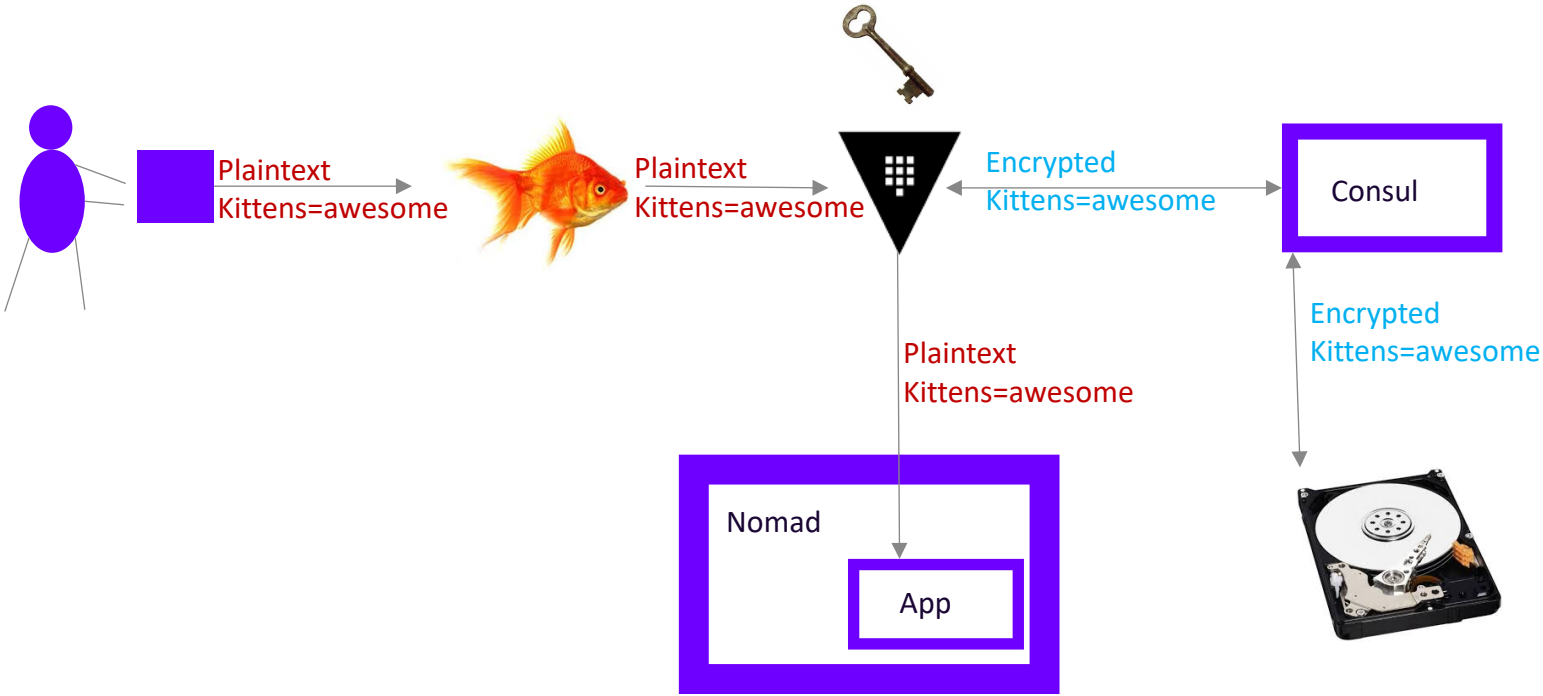
Apache Environment

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

Storage


Plaintext
Kittens=awesome
Plaintext
Kittens=awesome
Encrypted
Kittens=awesome
Consul
Encrypted
Kittens=awesome
Plaintext
Kittens=awesome
Nomad
App

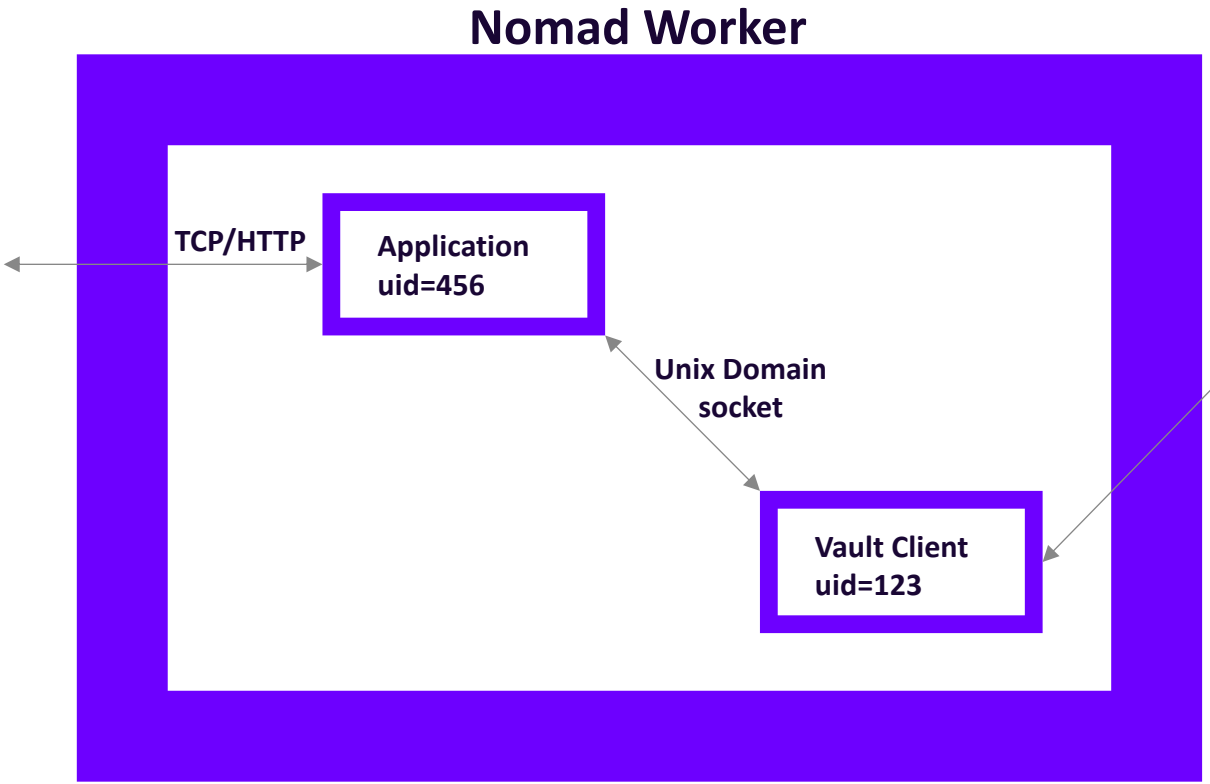
Demo
Deploy an application

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
- ☺

Demo
Cannon Fodder

The Future! / Questions?



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