Consul

Securing Cloud Native Communication:

From End User to Service

Daniel Bryant
Product Architect, Datawire

Nic Jackson
Developer Advocate, HashiCorp

Traditional IT approach to network security

tl;dr

Security is everyone’s responsibility

Application modernisation leads to heterogeneous infra/networks

Defence in depth is vital: edge/service comms security is one part of this

Mind the gap(s)!

All security must have good UX / DevEx

Who are we?

Nic Jackson

Developer Advocate, HashiCorp

@sheriffjackson

Daniel Bryant

Product Architect, Datawire

@danielbryantuk

Security is everyone’s responsibility

So, we don’t want to scare you, but…

214

Records containing personal data are exploited every second

So, we don’t want to scare you, but…

$3,860,000

Is the average cost of a data breach

So, we don’t want to scare you, but…

$350 0

techFT

Tech news and analysis from around the world

BA hit by biggest GDPR fine so far

By Chris Nuttall in London
July 7, 2019

British Airways has suffered the biggest fine yet levied under the EU's General Data P ion Regulation (GDPR), introduced in May last year to protect consumers' privacy an onal information.

The UK Information Commissioner's Office says it intends to fine BA £183m (€204m, $229m) — 1.5 per cent of BA's worldwide turnover in 2017 — after it admitted that more than half a million customers' data had been stolen by hackers last August from its website and mobile app.

WIRED

$700 Million Equifax Fine Is Still Too Little, Too Late

BUSINESS CULTURE GEAR IDEAS SCIENCE

SHARE

SHARE
TWEET
COMMENT
EMAIL

LILY HAY NEWMAN SECURITY 07.22.19 03:58 PM

$700 MILLION EQUIFAX FINE IS STILL TOO LITTLE, TOO LATE

TAMI CHAPPELL/REUTERS

TWO YEARS AFTER its historic data breach, the credit bureau Equifax agreed Monday to pay at least $575 million, and up to $700 million, to settle enforcement actions with 50 US

So, we don’t want to scare you, but…

72%

Increase in attacks between 2017 and 2018

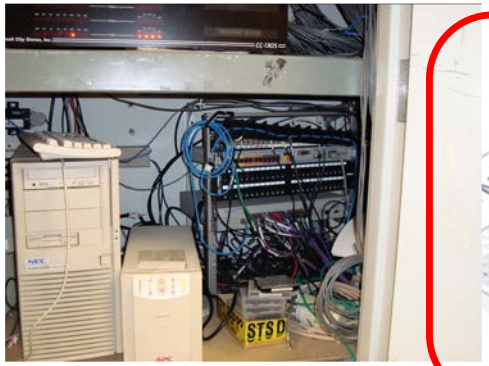
Gemalto Breach Level Index:

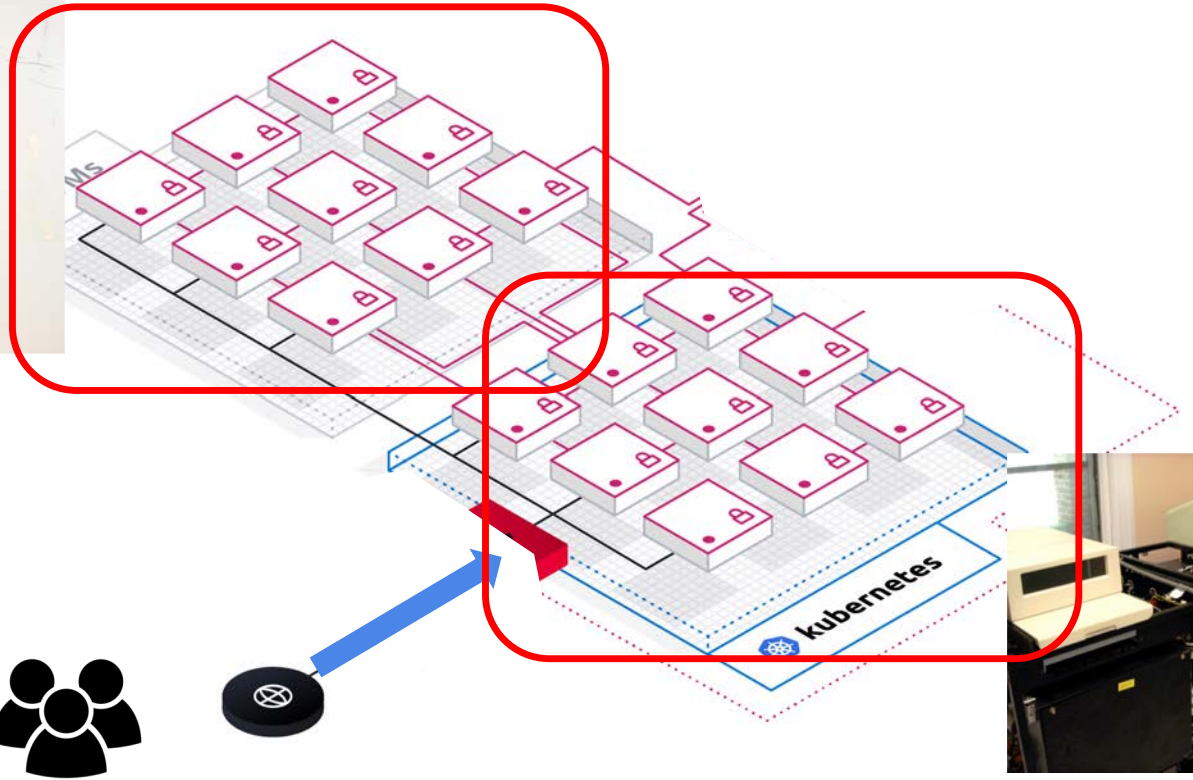
https://breachlevelindex.com/

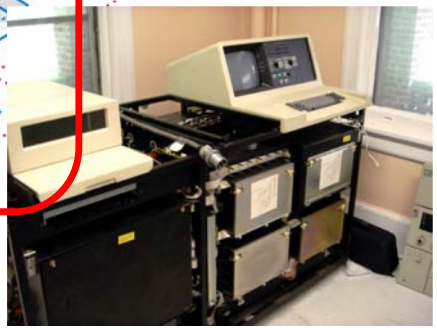
IBM Cost of a Data Breach Study:

https://www.ibm.com/security/data-breach

Application modernisation: Gift and curse







Defence in depth

Defence in depth is vital

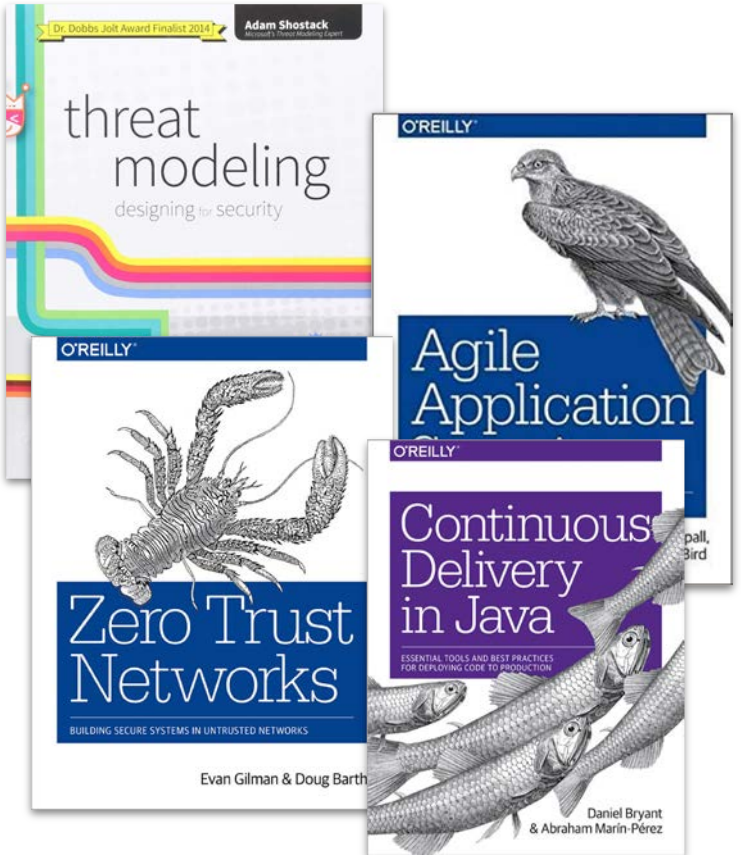
Harden and scan infrastructure

Scan code, dependencies, packages

Encrypt data at rest

Encrypt data in transit

Principle of least privilege



Defence in depth is vital

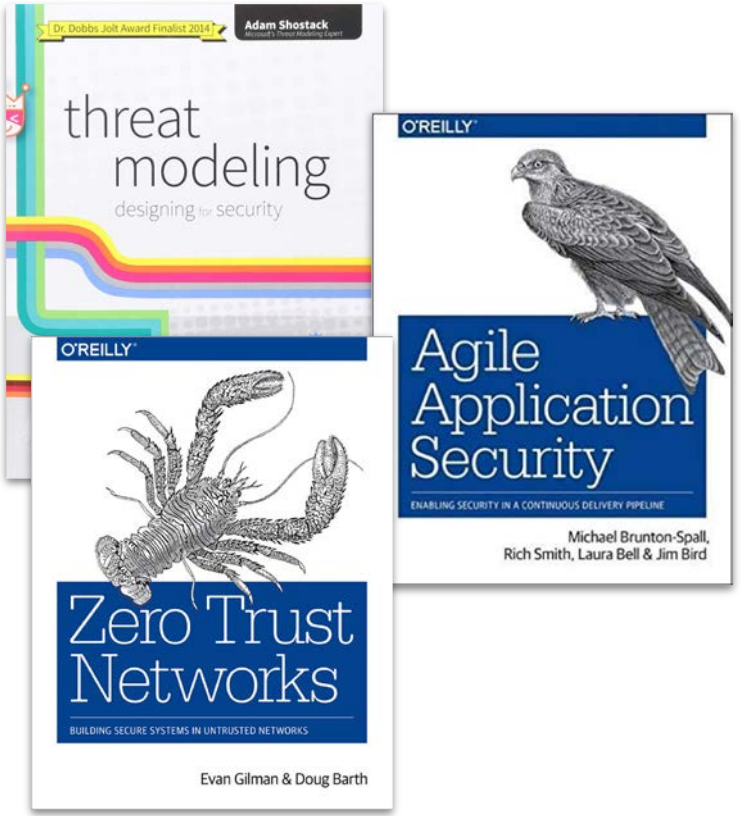
Harden and scan infrastructure

Scan code, dependencies, packages

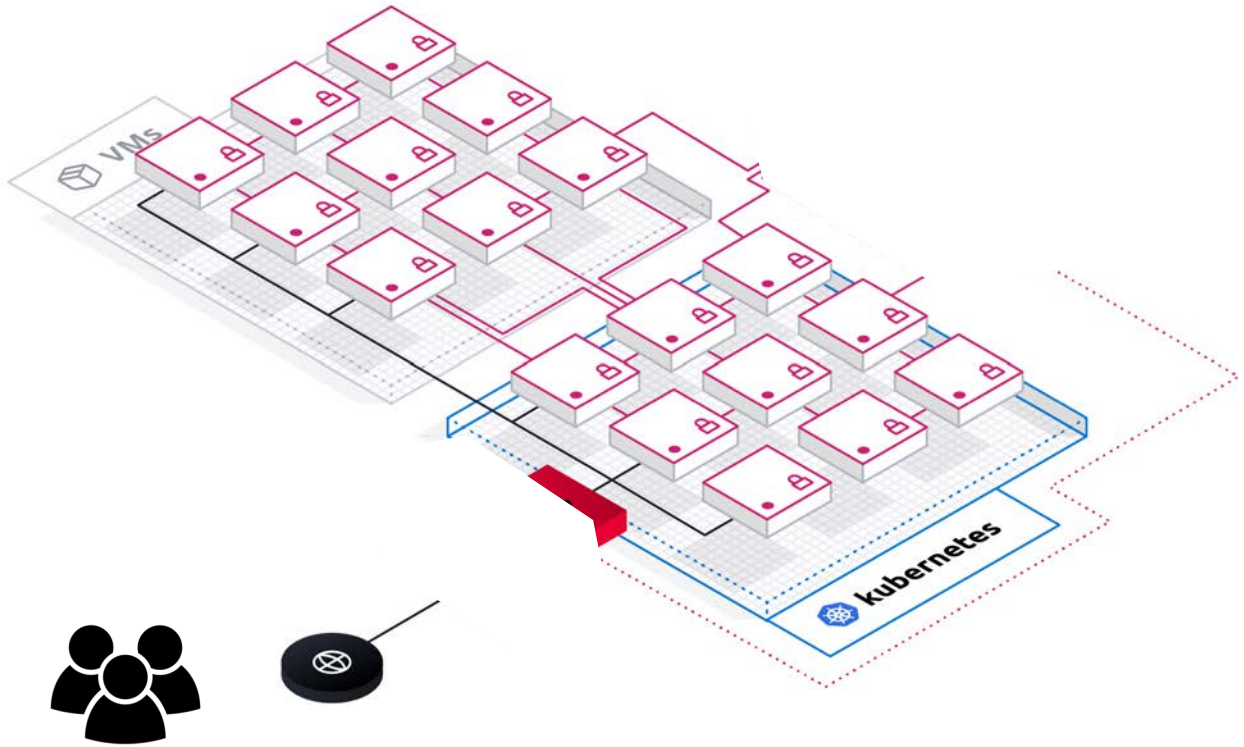
Encrypt data at rest

Encrypt data in transit

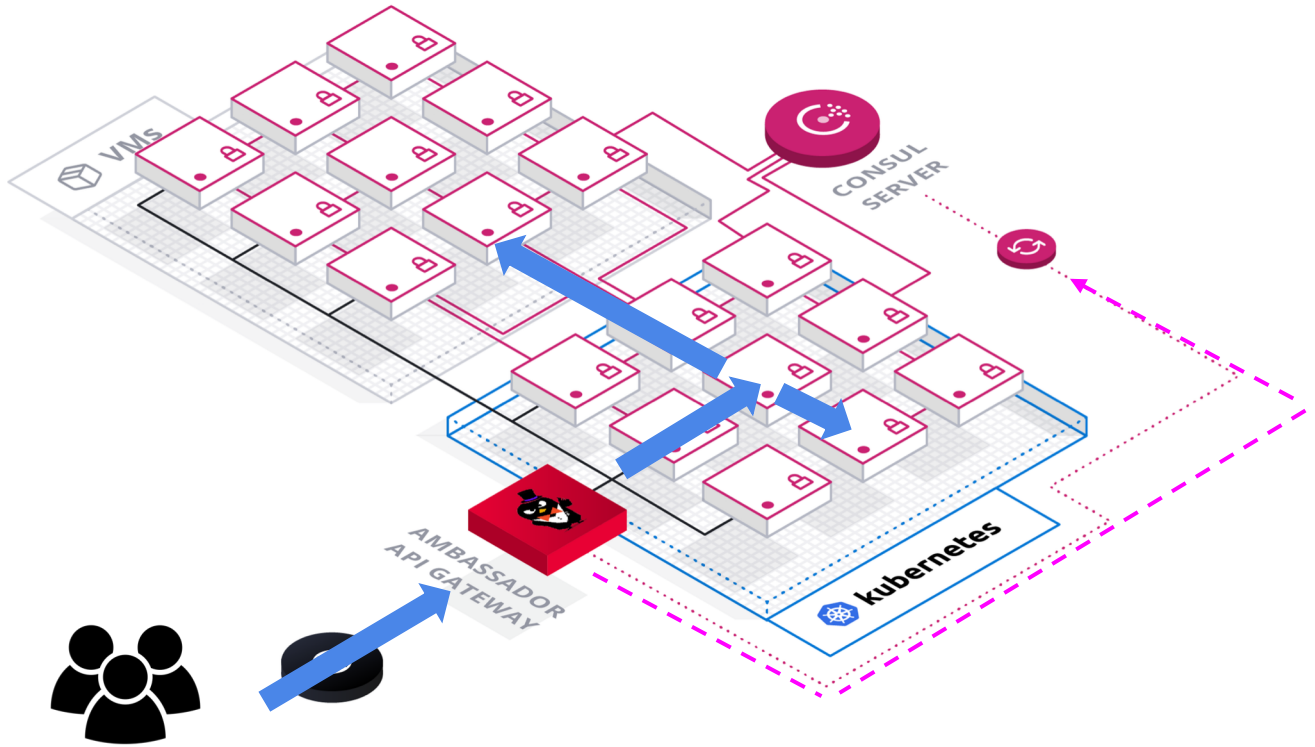
Principle of least privilege



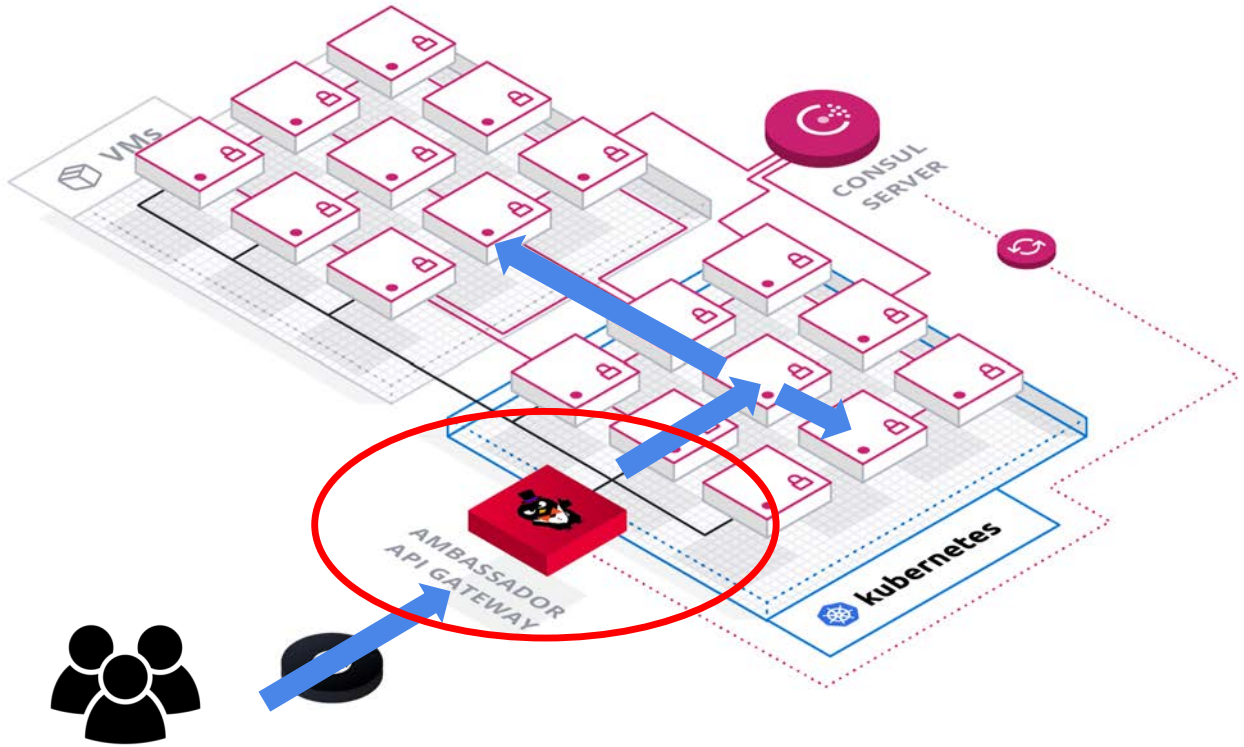
Exploring end-to-end communication



Exploring end-to-end communication



Exploring end-to-end communication



API Gateway: *Edge proxy, ingress, ADC…*

Exposes internal services to end-users (via multiple domains)

Encapsulates backends: k8s, VMs, bare metal etc

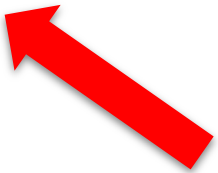
TLS termination: enforcing minimum TLS version

End-user authentication/authorization (add token/JWT for propagation)

Rate limiting: DDoS protection, etc

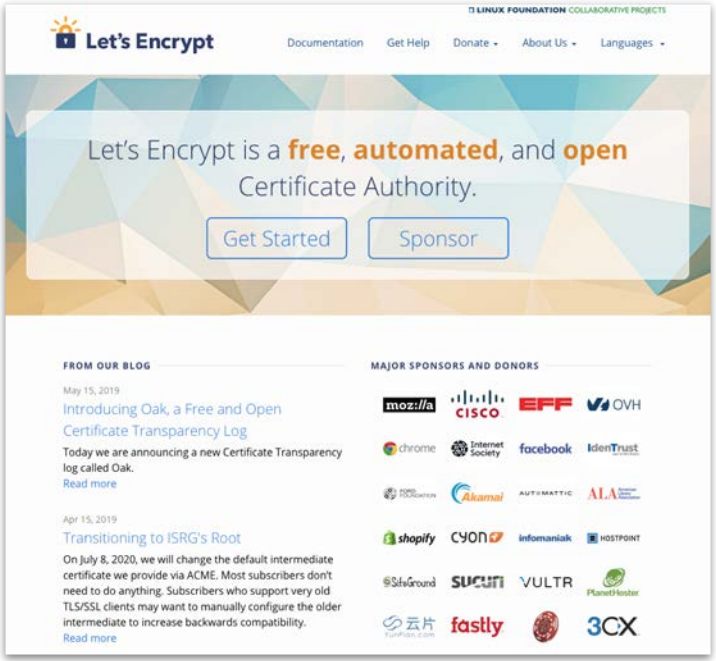
Ambassador config

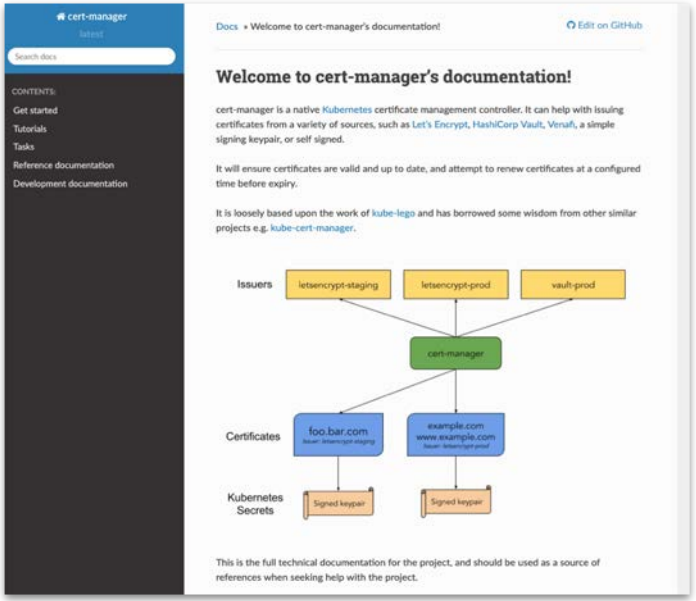
```yaml
---
apiVersion: getambassador.io/v1
kind: Mapping
metadata:
  name: consul-api-mapping
  namespace: default
spec:
  prefix: /api/
  timeout_ms: 20000
  host: emojify.today
  service: emojify-api-sidecar-proxy
  resolver: consul-dc1
  tls: ambassador-consul
  load_balancer:
    policy: round_robin
```



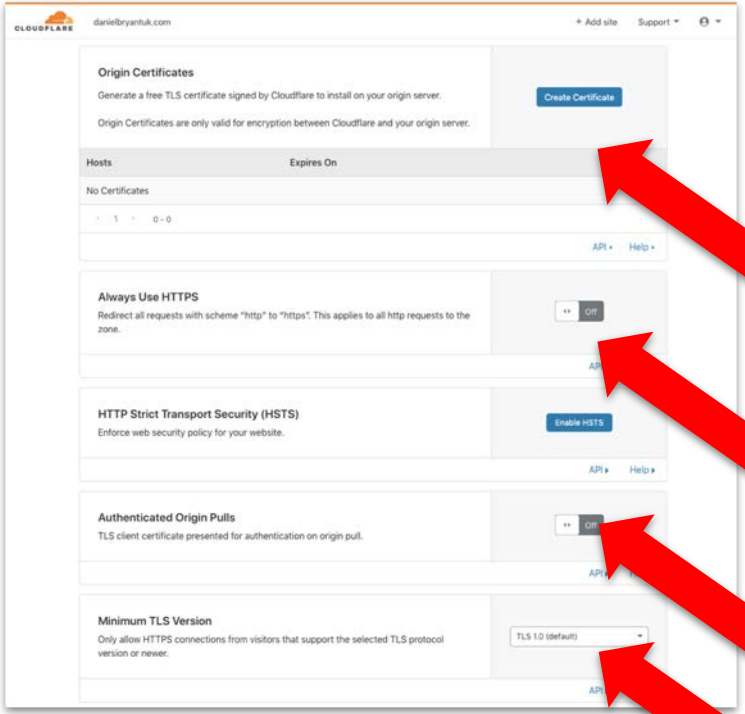

```yaml
---
apiVersion: getambassador.io/v1
kind: Module
metadata:
  name: tls
  namespace: default
spec:
  config:
    server:
      enabled: true
      secret: ambassador-certs
      redirect_cleartext_from: 8080
```

Friends don’t let friends manually issue TLS certs…



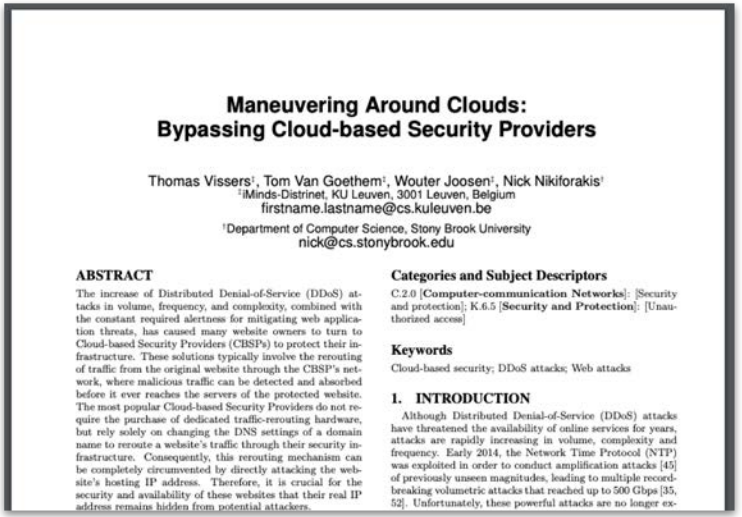


Quick Aside: CDNs



http://bit.ly/2JA0UAh

https://www.securitee.org/files/cloudpiercer_ccs2015.pdf



Maneuvering Around Clouds: Bypassing Cloud-based Security Providers

Thomas Vissers[‡], Tom Van Goethem[‡], Wouter Joosen[‡], Nick Nikiforakis[†]
[‡]iMinds-Distrinet, KU Leuven, 3001 Leuven, Belgium
firstname.lastname@cs.kuleuven.be
[†]Department of Computer Science, Stony Brook University
nick@cs.stonybrook.edu

ABSTRACT

The increase of Distributed Denial-of-Service (DDoS) attacks in volume, frequency, and complexity, combined with the constant required alertness for mitigating web application threats, has caused many website owners to turn to Cloud-based Security Providers (CBSPs) to protect their infrastructure. These solutions typically involve the rerouting of traffic from the original website through the CBSP's network, where malicious traffic can be detected and absorbed before it ever reaches the servers of the protected website. The most popular Cloud-based Security Providers do not require the purchase of dedicated traffic-rerouting hardware, but rely solely on changing the DNS settings of a domain name to reroute a website's traffic through their security infrastructure. Consequently, this rerouting mechanism can be completely circumvented by directly attacking the website's hosting IP address. Therefore, it is crucial for the security and availability of these websites that their real IP address remains hidden from potential attackers.

Categories and Subject Descriptors

C.2.0 [**Computer-communication Networks**]: [Security and protection]; K.6.5 [**Security and Protection**]: [Unauthorized access]

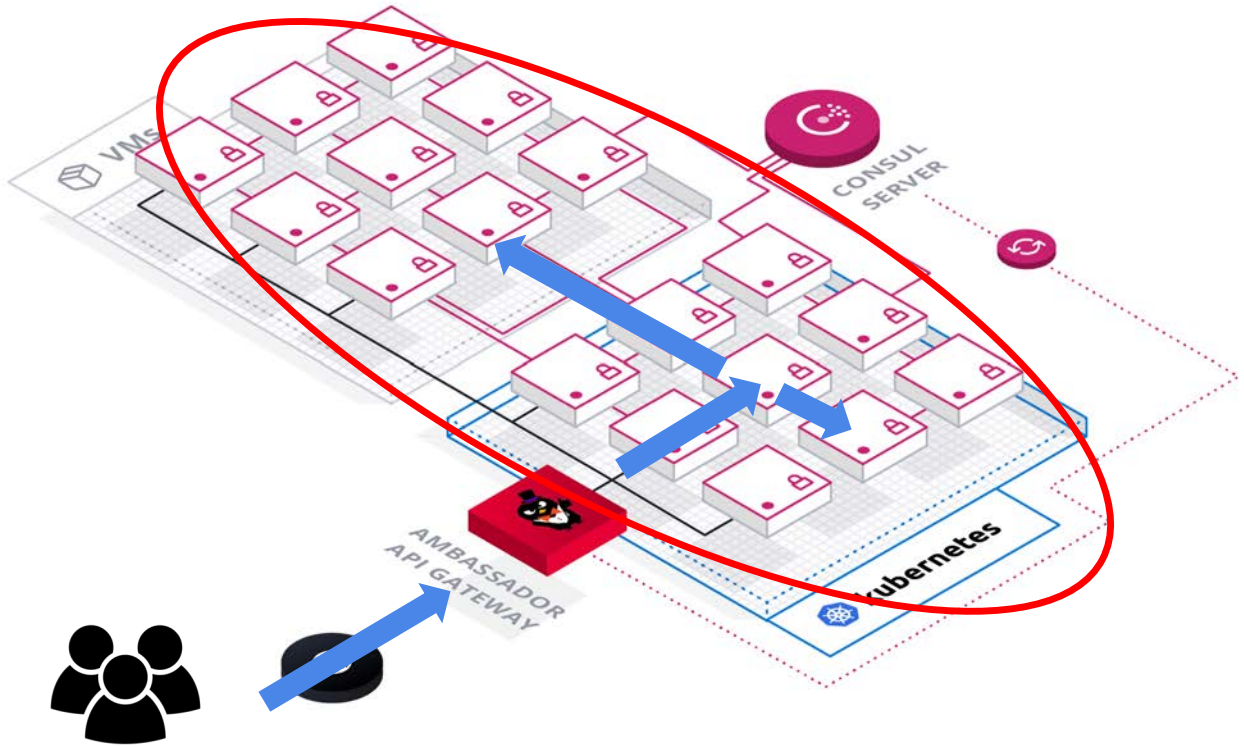
Keywords

Cloud-based security; DDoS attacks; Web attacks

1. INTRODUCTION

Although Distributed Denial-of-Service (DDoS) attacks have threatened the availability of online services for years, attacks are rapidly increasing in volume, complexity and frequency. Early 2014, the Network Time Protocol (NTP) was exploited in order to conduct amplification attacks [45] of previously unseen magnitudes, leading to multiple record-breaking volumetric attacks that reached up to 500 Gbps [35, 52]. Unfortunately, these powerful attacks are no longer ex-

Exploring end-to-end communication



Service Mesh: *Proxy mesh, Fabric model…*

Exposes internal services to internal consumers

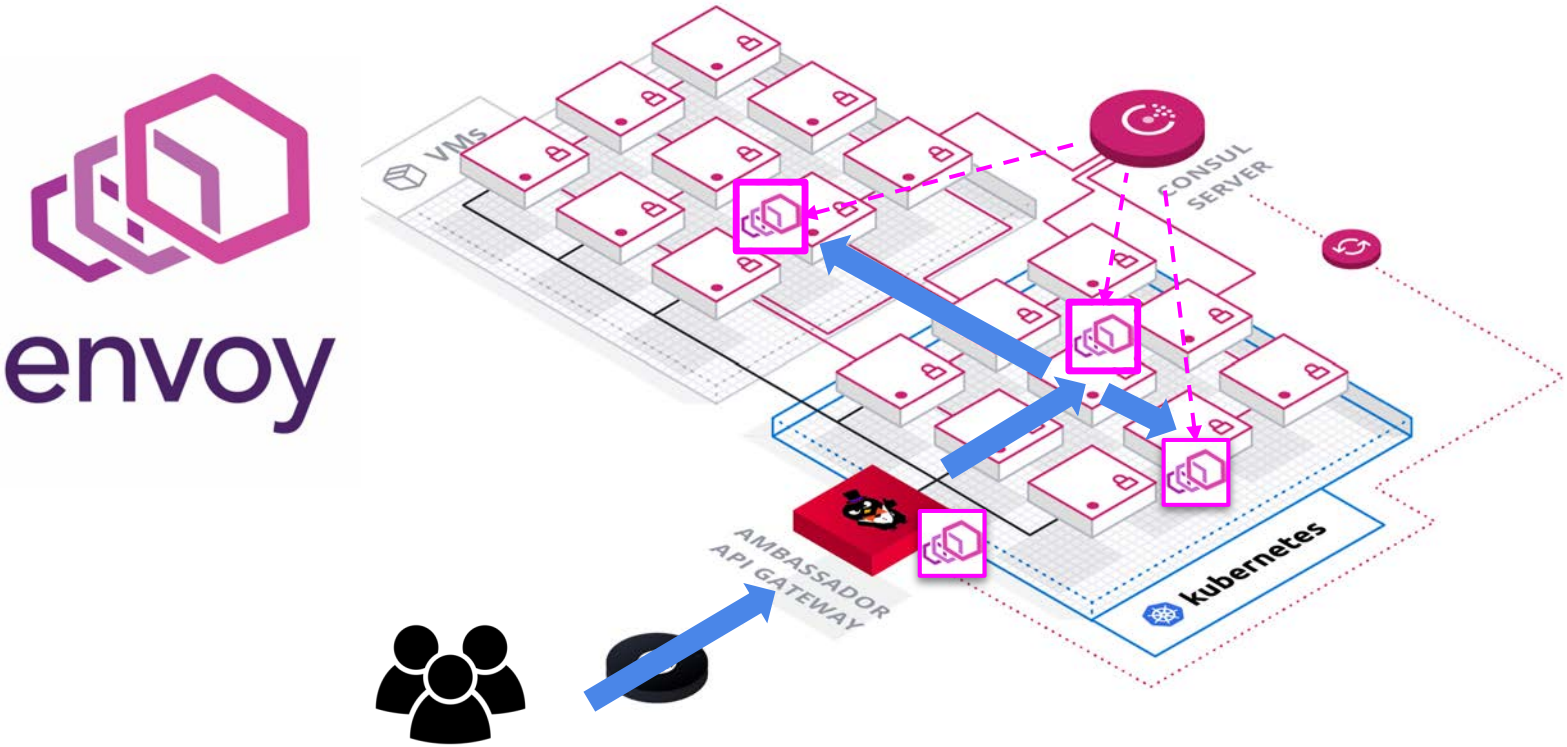
Encapsulates service infra: across k8s, VMs, bare metal etc

mTLS: service identity and traffic encryption

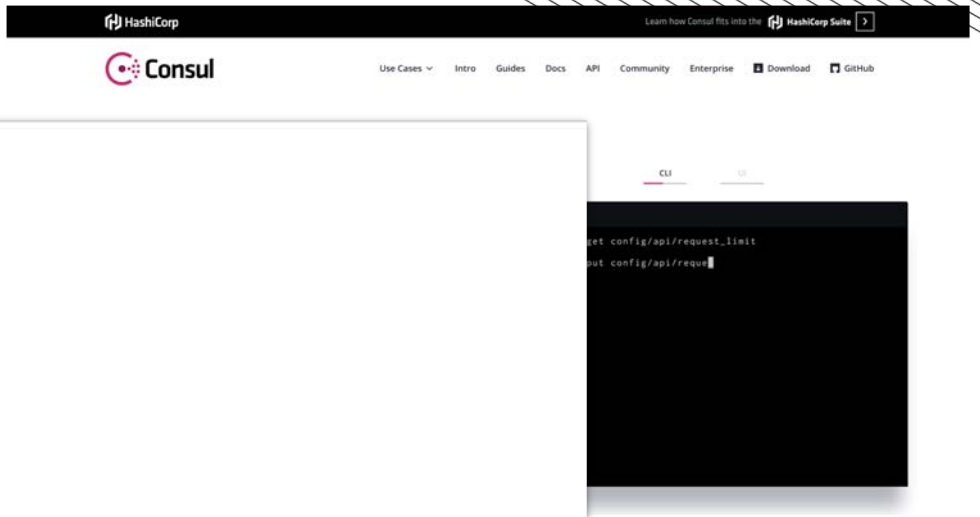
ACLs and intentions: infra/service identity-based access

Enforce metadata (but apps need to propagate headers/tokens)

Exploring end-to-end communication



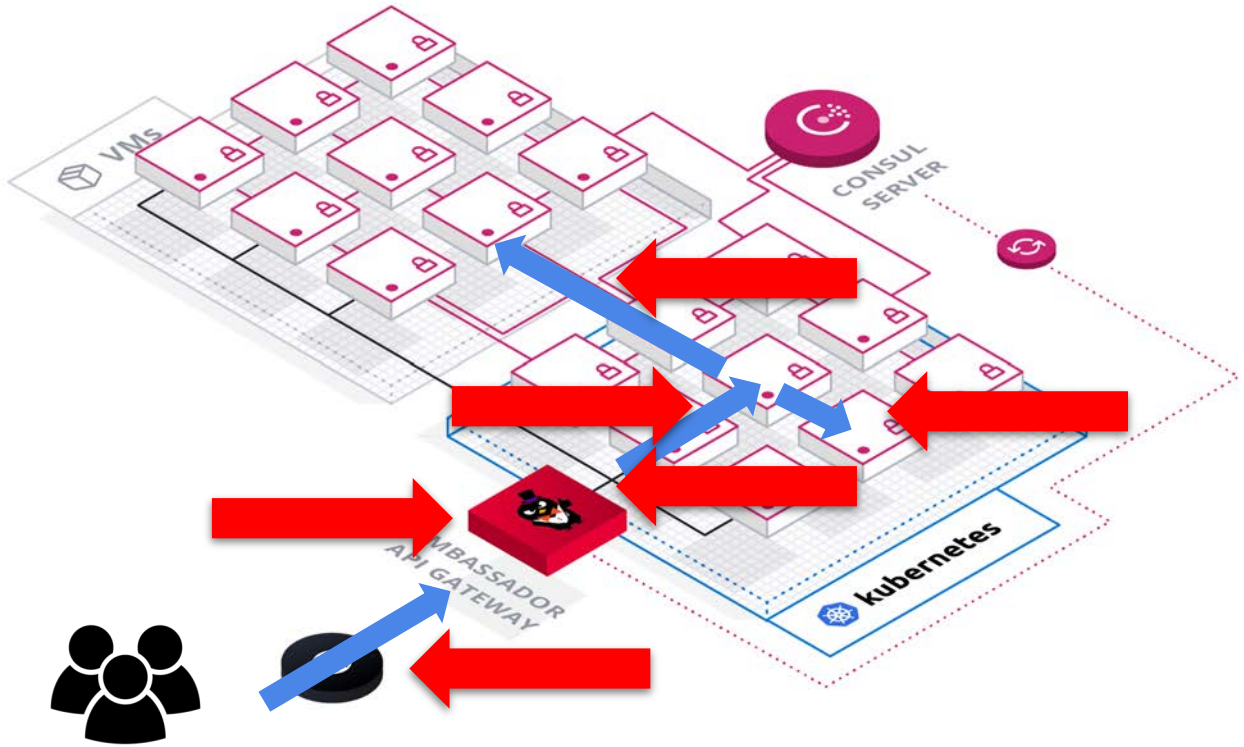
Consul config




```yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: emojify-api
  labels:
    app: emojify-api
spec:
  replicas: 1
  selector:
    matchLabels:
      app: emojify-api
  template:
    metadata:
      labels:
        app: emojify-api
      annotations:
        "consul.hashicorp.com/connect-inject": "true"
        "consul.hashicorp.com/connect-service-protocol": "http"
        "consul.hashicorp.com/connect-service-upstreams": "emojify-facedetect:8003,emojify-cache:8005"
        "prometheus_io_scrape": "true"
    spec:
```

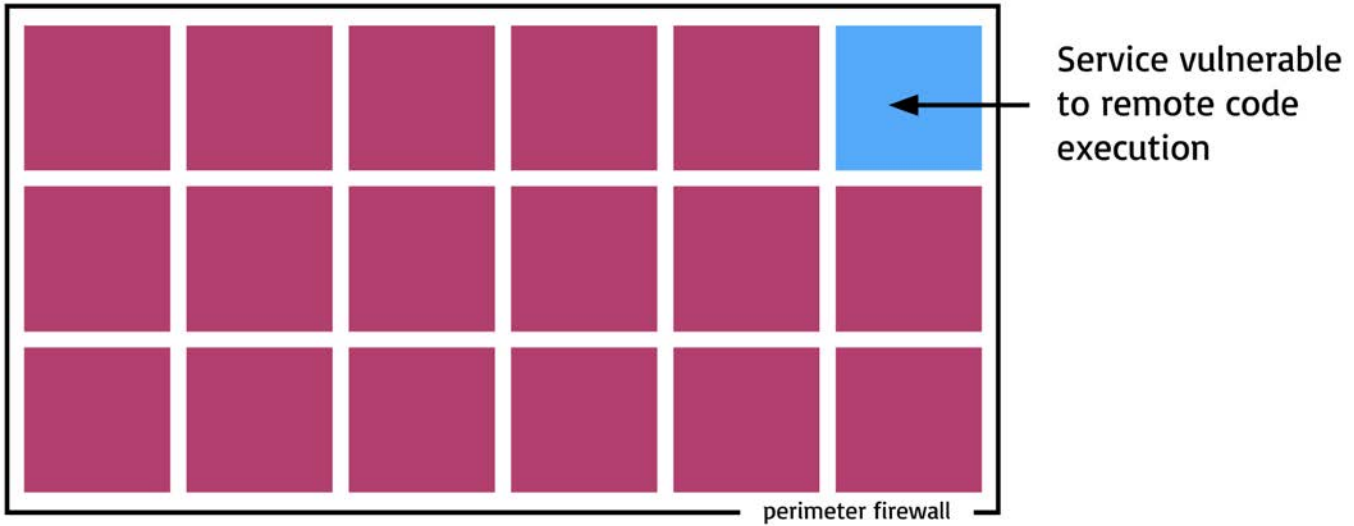

MIND THE GAP
http://flickr.com/carlez

Exploring end-to-end communication

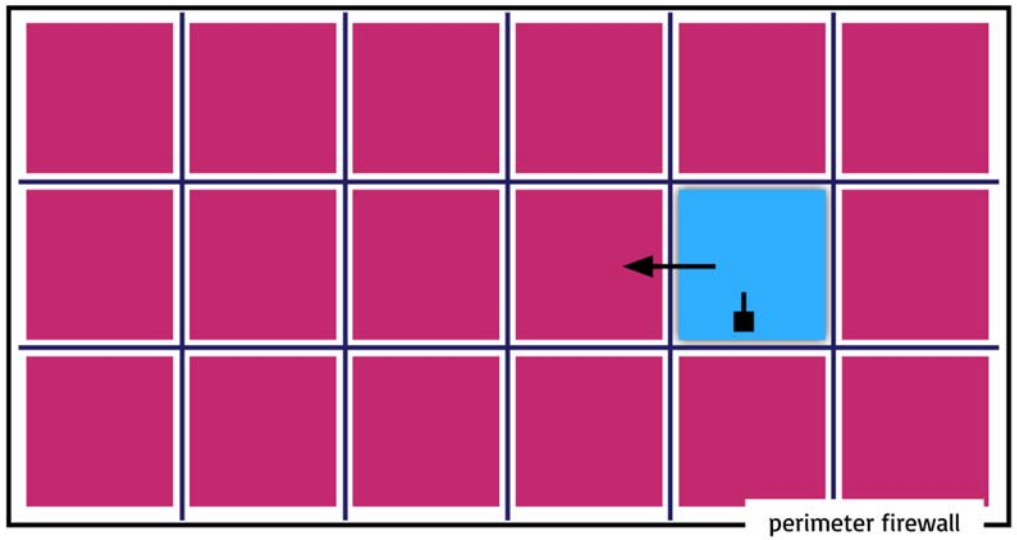


Identity and network segmentation

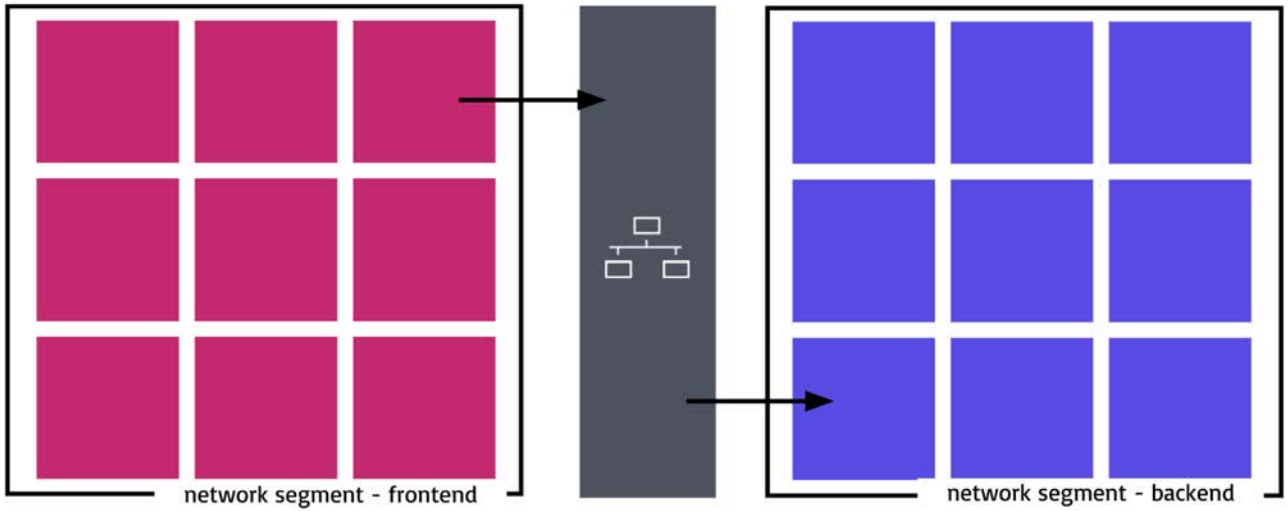
Bypass the perimeter by attacking services



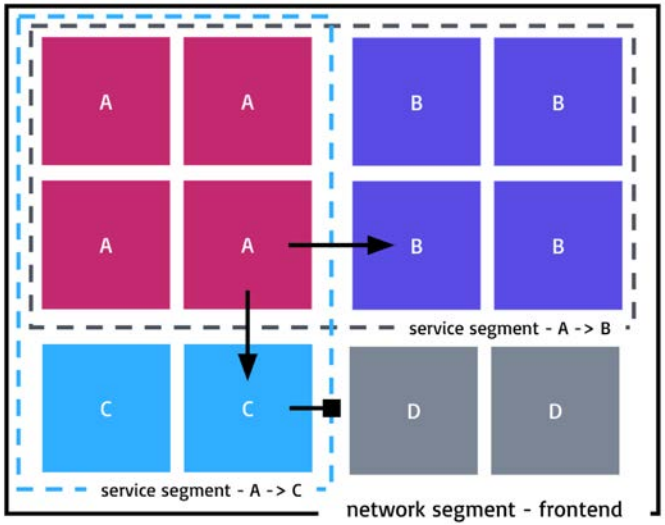
We need internal network isolation



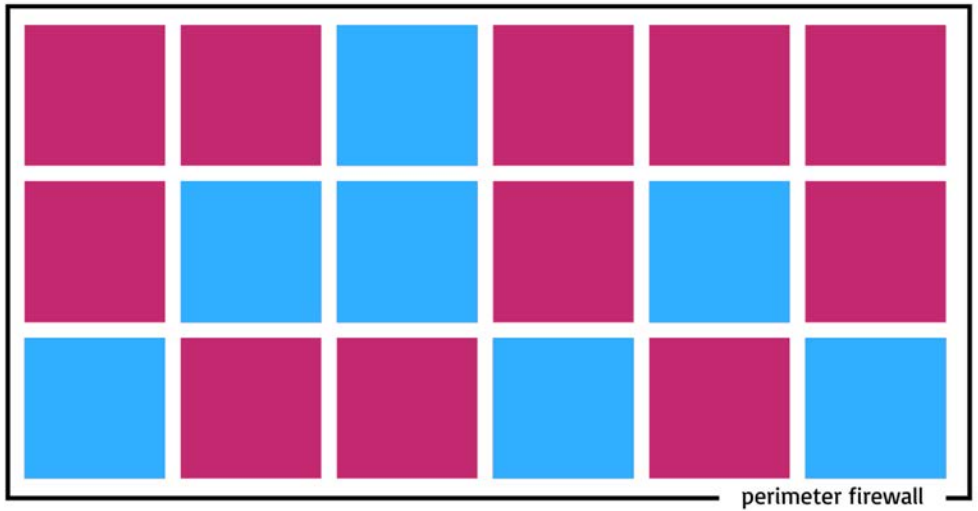
Network segmentation



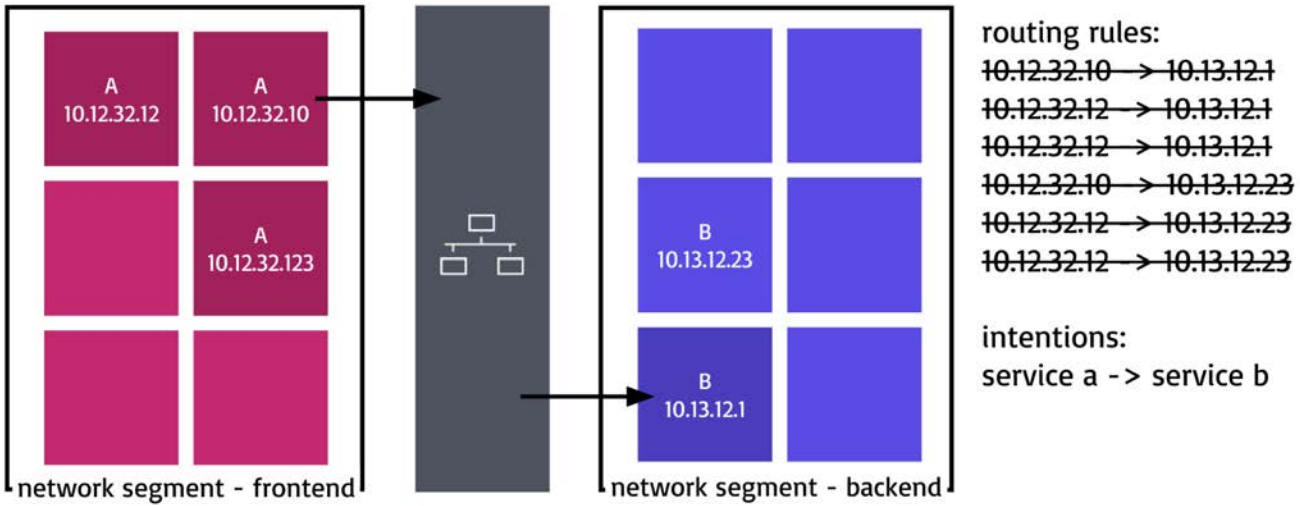
Service segmentation



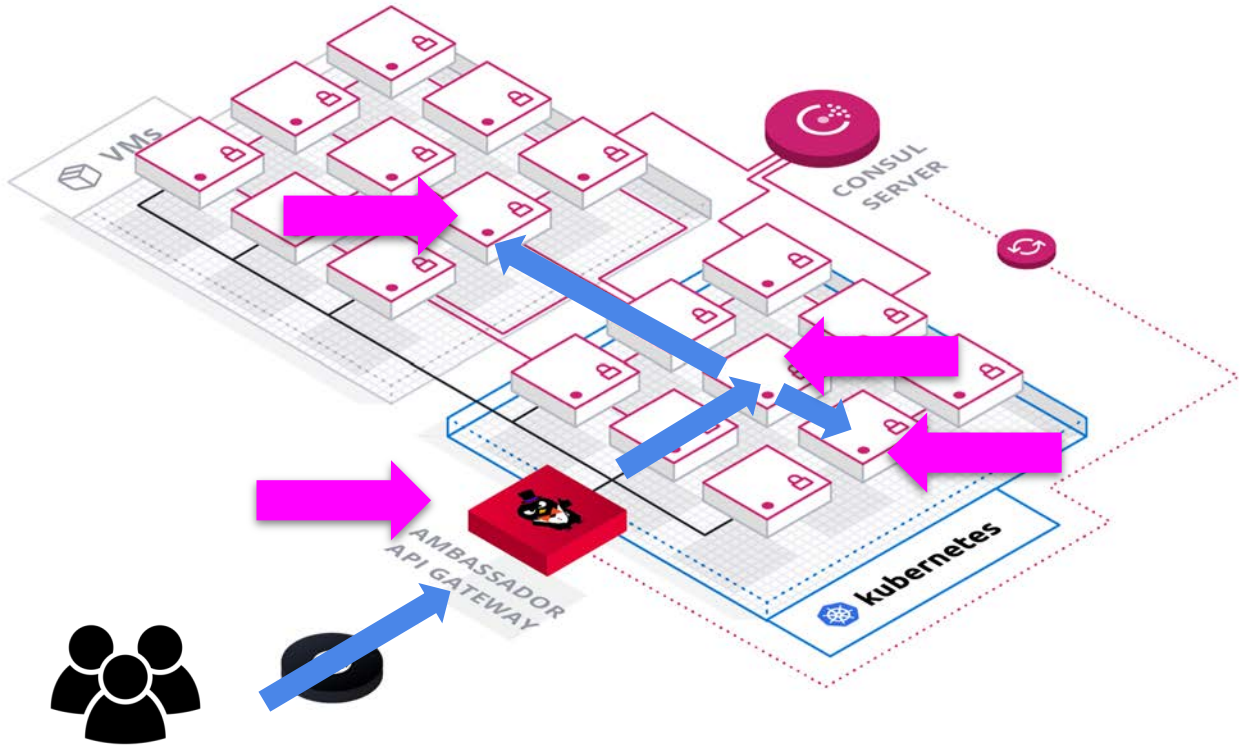
Problem: Dynamic environments...



Network / Service segmentation with intention-based security



Exploring end-to-end communication



Consul config

HashiCorp — Learn how Consul fits into the HashiCorp Suite

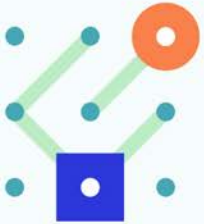
Consul — Use Cases | Intro | Guides | Docs | API | Community | Enterprise | Download | GitHub

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: emojify-api
  labels:
    app: emojify-api
spec:
```

```
$ consul intention create -d
Created: web => db (deny)
```

```
---
kind: TrafficTarget
apiVersion: access.smi-spec.io/v1alpha1
metadata:
  name: emojify-website-targets
  namespace: default
destination:
  kind: ServiceAccount
  name: emojify-website
  namespace: default
sources:
- kind: ServiceAccount
  name: ambassador
  namespace: default
specs:
- kind: TCPRoute
  name: emojify-website-tcp-route
```

Service Mesh Interface



A standard interface for service meshes on Kubernetes.

Consul

Demo

Conclusion

Security is everyone’s responsibility

Application modernisation leads to heterogeneous infra/networks

Defence in depth is vital: edge/service comms security is one part of this

Mind the gap(s)!

All security must have good UX / DevEx

References

Context:

- https://www.infoq.com/articles/api-gateway-service-mesh-app-modernisation/

Reference:

- https://www.getambassador.io/user-guide/consul-connect-ambassador/
- https://www.getambassador.io/user-guide/consul/
- https://www.consul.io/docs/platform/k8s/ambassador.html
- https://www.hashicorp.com/blog/hashicorp-consul-supports-microsoft-s-new-service-mesh-framework

Experiment in an Instruqt sandbox: https://instruqt.com/hashicorp/tracks/sock-shop-tutorial

Code examples: https://github.com/emojify-app

Consul

Questions?

Consul

Thanks!

@sheriffjackson | @danielbryantuk

Consul

Bonus

Service Mesh: Three Pillars

Observability

- “Golden signals”: latency, errors, traffic, saturation (USE, RED)
- Both global and service-to-service

Reliability

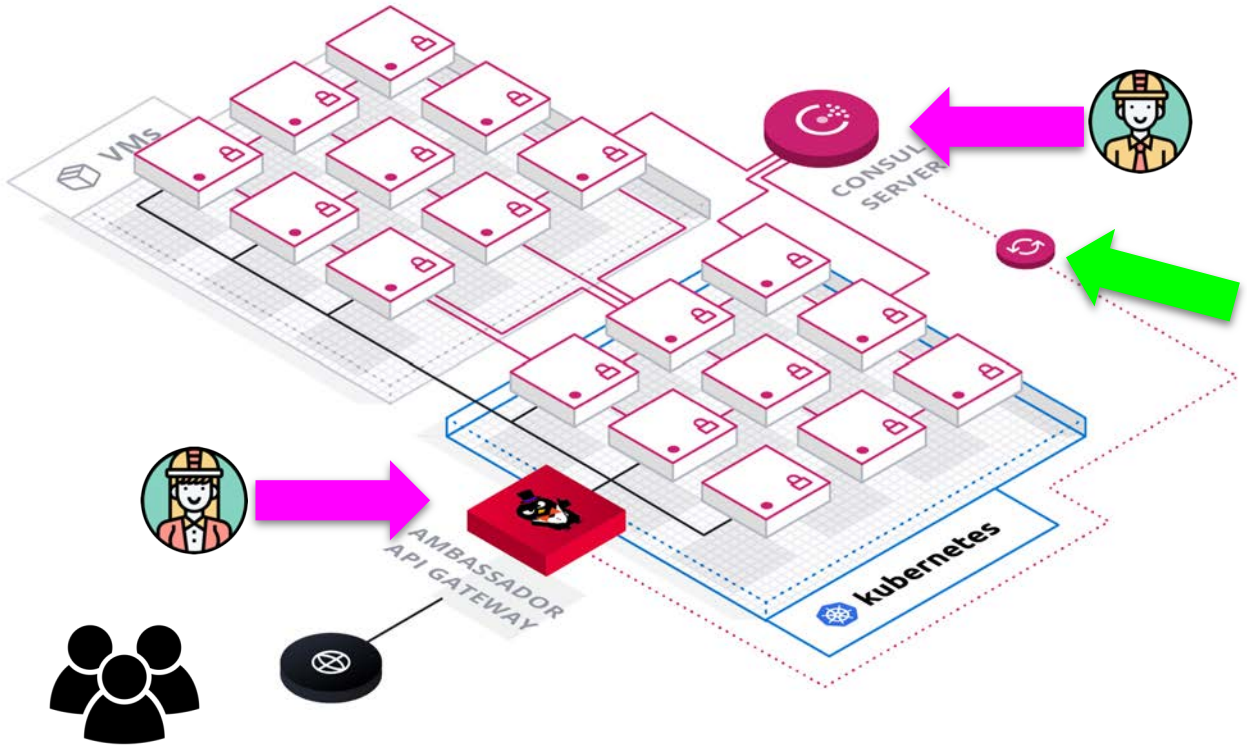
- Abstracting health checks, retries, circuit breakers etc.
- Providing sane default to protect system

Security

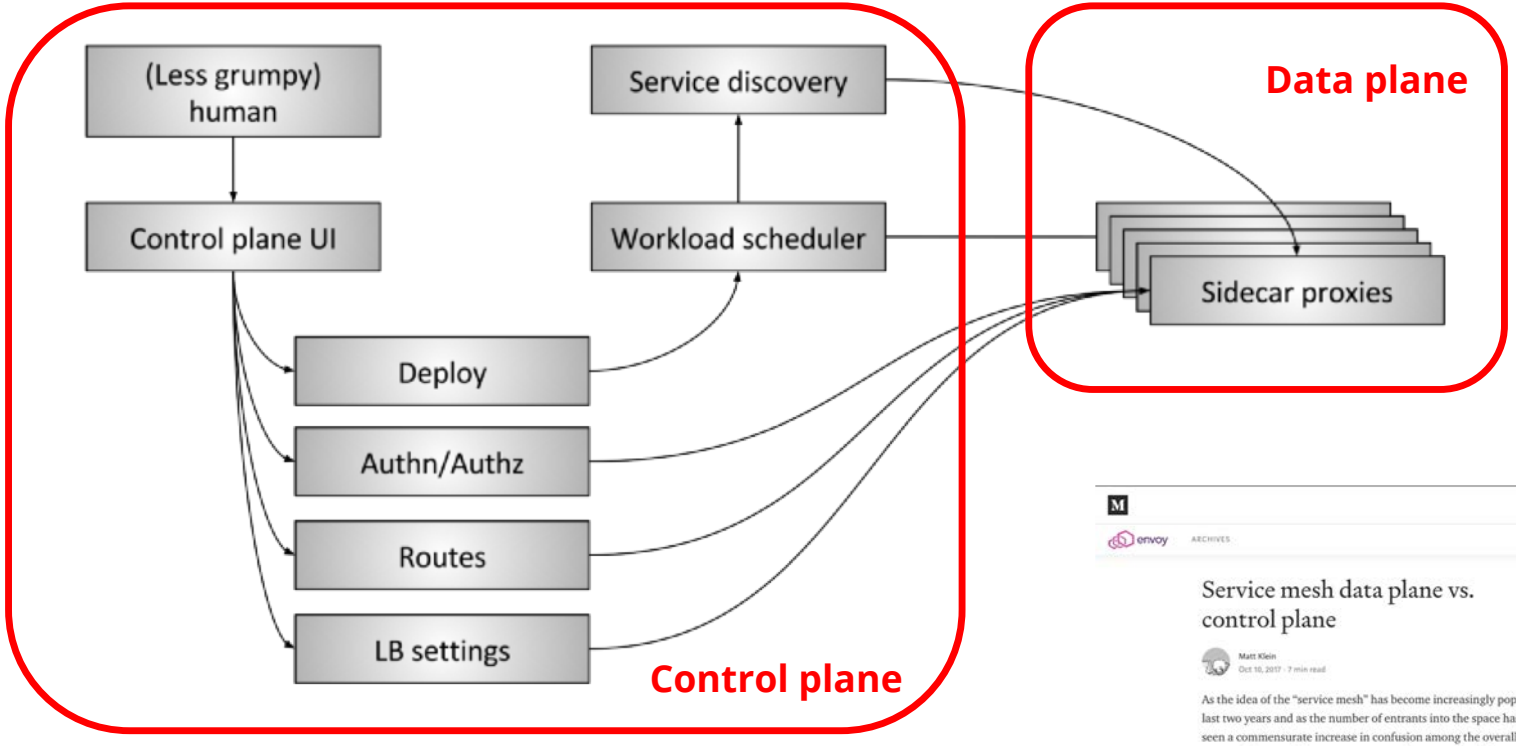
- Authn/z propagation, mTLS, network segmentation

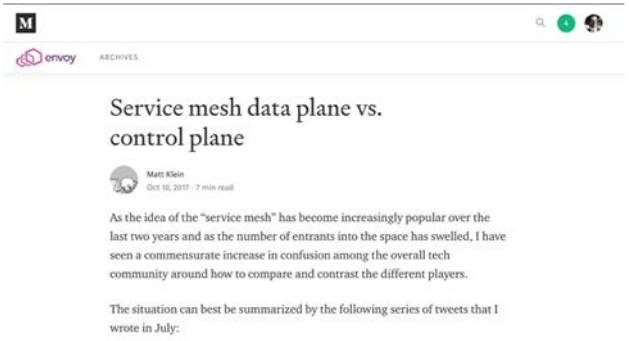
Security must have good UX

Exploring end-to-end communication



Control planes and data planes




envoy ARCHIVES

Service mesh data plane vs. control plane

Matt Klein
Oct 10, 2017 · 7 min read

As the idea of the "service mesh" has become increasingly popular over the last two years and as the number of entrants into the space has swelled, I have seen a commensurate increase in confusion among the overall tech community around how to compare and contrast the different players.

The situation can best be summarized by the following series of tweets that I wrote in July:

https://blog.envoyproxy.io/service-mesh-data-plane-vs-control-plane-2774e720f7fc

Control planes: Differing use cases

North-south

- Unknown / untrusted clients
- Limited exposure of services (Mapping)
- Centralised ops ingress defaults + decentralised product team cfg

East-west

- Dynamic service information update required (multiple sources)
- Identity required for all services (mTLS + ACLs)
- “Sane” internal defaults + decentralised dev cfg

Ambassador + Consul