

Get your APIs Secured  
with Otoroshi !



**Pass The SALT 2019**



Mathieu Ancelin  
@TrevorReznik  
SERLI



Chris Woodrow  
@StrangeCousin  
MAIF



## **Disclaimer**

**We are not security experts ! We are here to share our experience about securing APIs, the problems we've been facing and the solutions we've been providing.**



# 2016 : We had to move ... Quick



French mutual insurance company  
Est. in 1934

In 2016 we needed to move :

- Connected cars
- Connected homes
- Connected ...

Connected *whatever* == Less Risk ==  
Less Insurance

Our Strategy : diversification



# Let's build a new platform !





# One goal : get focused on business value

We believe we need a good  
technical basis to make good  
business

Making developers' life easy (and  
get as close as possible to the  
state of the art) :

- (Clever) Cloud hosting
  - Scalability
  - Resiliency
  - ...
- Automation (CI/CD)
- Metrics
- Monitoring
- ...



One of our first thoughts :  
“How to manage APIs ?”

Let's build our own stuff !!



# Here comes ... Otoroshi

API management on top of a  
reverse proxy

<https://maif.github.io/>





# Typical Features



Exposing APIs



Quotas and throttling



**I'LL BITE YOUR LEGS OFF!**

Resiliency



Monitoring



And, of course ...

# Security





# Exchange protocol



Otoroshi is a reverse proxy :

- Apps need to make sure that requests actually come from Otoroshi
- Otoroshi needs to know if apps are proxied despite themselves



# Descending exchange protocol

Otoroshi -> Apps

Otoroshi sends a header containing a signed random state with a short TTL is sent with the request to the proxied app which validates signature

```
curl -H 'Otoroshi-State:  
eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJzdGF0ZSI6I  
jEYMzQ1Njc4OTAiLCJpYXQiOiE1NjE2NDA1MTEsImV4cCI6MTU  
2MTY0MDUyMX0.1-ky2uu1FRQwSyxAL_VZH9Ju_98gb19uR41xF  
3aEnSNaSrvovpY_cY7ekoWKUdThJnbV1cCC7VaUBVF8UvQE8w'  
https://api-backend.foo.bar:8443 --include
```



# Ascending exchange protocol

Otoroshi <- Apps

Application sends back a header containing the original random state, signed and with a short TTL in the response to Otoroshi

```
HTTP/1.1 200 OK
Date: Thu, 27 Jun 2019 13:05:01 GMT
Otoroshi-State-Resp:
eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJzdGF0ZS1yZS1uIjoiMTIzNDU2Nzg5MCI6Im1hdCI6MTU2MTY0MDUxNSwiZXhwIjoxNTYxNjQwNTI1fQ.4kRPy01tybeMamMdoGGkQjLC-4k0-oj_EdOPItmYEE_YW40D1N9LfdSt02hMC4-VqohrrMvQlRtlycR3Ss0qWg
Content-Type: application/json
Content-Length: 20
```

```
{"msg": "Hello World!"}
```



# TLS

## Otoroshi handles TLS dynamically

- Server-side
  - may be enforced
- Client-side
  - may be enforced
- End-to-end (m)TLS
  - from client to otoroshi, from otoroshi to app
  - no passthrough
    - at least for HTTP routing
- Authority Validation
  - Local Authority
    - Trust Store
  - External Authority
    - check if the service is authorised for the current user with the the current certificate chain



# JWT tokens

JWT tokens are a very common way to provide contextual or authentication / authorization information in APIs world

JWT tokens validation :

- shared secret (HMAC)
- pub/priv keys (RSA, ECDSA)

JWT tokens remediation

- fields value validation
- token location changes
- fields transformation
- sign with another secret / keypair



# Webapps security

Otoroshi provides login as a service for any proxied web app that can integrate with authn services :

- OAuth, OIDC, LDAP, Local, etc.

Can add security headers :

- Strict-Transport-Security, Public-Key-Pins, X-Permitted-Cross-Domain-Policies, X-XSS-Protection, Referrer-Policy, Content-Security-Policy, X-Content-Type-Options, X-Frame-Option, etc.



# APIs Security

## Api Keys for Machine to Machine calls

- Classical
  - Client ID + Client Secret
  - Basic Auth
- Token
  - Signed
  - Request parts
- Third party Api Key validation
  - for instance OIDC
  - using an external IAM

## Api Keys can be constrained

- Token TTL
- Routing based on metadata



# Misc.

A few more security features :

- Headers validation
- IP addresses blacklist / whitelist
- Auditing
- FIDO U2F support for backoffice



# Nice UI API-drivable

swagger  Explore

## Otoroshi Admin API <sup>1.0.3</sup>

[ Base URL: otoroshi-api.foo.bar/api ]  
<http://otoroshi.foo.bar:8080/api/swagger.json>

Admin API of the Otoroshi reverse proxy

[Contact Otoroshi Team](#)  
[Apache 2.0](#)  
[Find out more about Otoroshi](#)

Schemes: HTTP

configuration  
import Everything

おとろし Otoroshi

SEARCH: Search service, line, etc ...

DASHBOARD

SERVICES

- All services
- For line experiments
- For line preprod
- For line dev
- For line sandbox
- For line prod
- + Add service
- + Add service from a CleverApp

otoroshi おとろし

LIVE METRICS

62.42 requests per second	16.35 ms. per request	5.18 ms. overhead per request
0 Mb/sec in	1.368 Mb/sec out	1 concurrent requests

GLOBAL METRICS

22 679 536 requests served	51.487 Gb in	814 Gb out
----------------------------	--------------	------------

MAIF



#OSSbyMAIF

Made with love

Otoroshi is Open  
Source since Jan.  
2018

5 minutes tutorial : <http://bit.ly/try-otoroshi>



We have more  
and more to  
come ...

Designing & building commons since 1934

Prix du Meilleur Projet Open Source 2018

LES ACTEURS DU LIBRE  
3ème édition

**melusine**  
Python  
Melusine is a high-level library for emails classification and feature extraction "dédiée aux courriels français".  
github.com/MAIF/melusine

**otoroshi**  
Scala  
Reverse proxy that handles traffic from and to your micro-services, with API management.  
maif.github.io/otoroshi

**izanami**  
Scala  
Shared configuration service for micro-services, provides feature flipping and A/B Testing.  
maif.github.io/izanami

<https://maif.github.io/>



Thank you !