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

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
- ...

https://maif.github.io/
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

Resiliency

Monitoring
And, of course ...
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

```bash
curl -H 'Otoroshi-State: eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJzdGF0ZSI6IjEyMzQ1Njc4OTAiLCJpYXQiOjE1NjE2NDA1MTEsImV4cCI6MTU2MTY0MDUyMX0.1-ky2uu1FRQwSyxAL_VZH9Ju_98gb19uR41xF3aEnSNsSrvoypY_cY7ekoWKUdThJnbV1cCC7VaUBVF8UvQE8w' https://api-backend.foo.bar:8443 --include
```
Ascending exchange protocol

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.eyJzdGF0ZSI6IjoxNTYxNjQwNTI1fQ.4kRPy01tybeMamMdoGGkQjLC-4k0-oj_EdOPItmYEE_YW4OD1N9LfdStO2hMC4-VqohrrMvQlRtlycR3Ss0qWg
Content-Type: application/json
Content-Length: 20

{"msg":"Hello World!"}
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 current certificate chain

https://maif.github.io/
JWT tokens are a very common way to provide contextual or authn/authz informations 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
A few more security features:

- Headers validation
- IP addresses blacklist / whitelist
- Auditing
- FIDO U2F support for backoffice
Nice UI
API-drivable
Otoroshi is Open Source since Jan. 2018

We have more and more to come ...

https://maif.github.io/
Thank you !