Universitat Oberta de Catalunya

uoc.edu

# Security in API and API managers

**Nombre del estudiante:** Bibián Castro Millán
Máster Interuniversitario en Seguridad de las TIC
Protocolos y aplicaciones de seguridad

**Nombre Consultor/a:** Manuel Jesus Mendoza Flores
**Profesor/a responsable de la asignatura:** Víctor Garcia Font

03/06/2019

# Agradecimientos

A mi abuelo por haberme inculcado la importancia de la formación académica.

A mi abuela por estar siempre apoyándome.

A mi madre por enseñarme a luchar.

**FICHA DEL TRABAJO FINAL**

| | |
|---|---|
| **Título del trabajo:** | *Security in API and API managers* |
| **Nombre del autor:** | *Bibián Castro Millán* |
| **Nombre del consultor/a:** | *Manuel Jesus Mendoza Flores* |
| **Nombre del PRA:** | *Víctor Garcia Font* |
| **Fecha de entrega (mm/aaaa):** | 06/2019 |
| **Titulación::** | *Máster Interuniversitario en Seguridad de las TIC* |
| **Área del Trabajo Final:** | *Protocolos y aplicaciones de seguridad* |
| **Idioma del trabajo:** | *Español* |
| **Palabras clave** | *API Management, API, REST* |


**Resumen del Trabajo (máximo 250 palabras):** *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El gran número de APIs existentes en la actualidad y las necesidades de interacción entre las distintas aplicaciones en Internet han provocado el surgimiento de los sistemas de API Management, cuyo objetivo es el de administrar de una manera fácil y centralizada las distintas tareas involucradas en el ciclo de vida de las APIs.

El principal objetivo de este trabajo de fin de máster es el estudio de los servicios de seguridad que ofertan los sistemas de API Management del mercado actual, para poder auditar que realmente estos sistemas se encuentran preparados para combatir las principales amenazas actuales de seguridad en el marco de las APIs y de aplicaciones web.

Gracias al estudio y análisis realizado de los servicios de seguridad de las distintas soluciones analizadas en el presente trabajo de fin de máster, se ha podido realizar una comparativa, desde el punto de vista de seguridad, que permite discernir de una manera rápida las debilidades y fortalezas de cada solución de API Management objeto de estudio, en las distintas funciones de seguridad ofertadas.

**Abstract (in English, 250 words or less):**

The large number of currently existing APIs and the interaction needs between different applications on the Internet have led to the emergence of API Management systems, whose objective is to manage in an easy and centralized form the different tasks involved in the APIs's life cycle.

The main objective of this master's thesis is to study the security services offered by the API Management systems of the current market, in order to be able to audit that these systems are really prepared to combat the main current security threats in the market related to APIs and web applications.

Due to the study and analysis carried out of the security services of the different solutions studied in this master's thesis, it has been possible to make a comparison, from the point of view of security, which allows us to quickly discern weaknesses and strengths of each API Management solution object of study in the different security functions offered.

# Índice

# Índice de figuras y tablas

# 1. Introducción

## 1.1. Contexto y justificación

En la actualidad la constante transformación digital de la sociedad y el auge de las TICs han derivado en multitud de servicios tecnológicos cuyo fin es el de aumentar la comodidad de las personas, ya sea posibilitando la realización de tareas de una manera más simple o automatizando las mismas. En este tecnológico escenario, Internet juega un papel fundamental, donde las empresas a menudo centralizan su oferta en forma de servicios web o aplicaciones.

Dada la multitud de servicios web existentes, tanto de índole privada como pública, surge la necesidad de comunicar estos servicios para que puedan consumir datos los unos de los otros o incluso aportarlos. Este fue uno de los motivos del nacimiento de las APIs, la comunicación de servicios. Actualmente existente millones de APIs asociadas a servicios y aplicaciones de distinta tipología: web, desktop, móviles…

Dentro de una misma empresa es lógico que existan varias APIs en función de los distintos departamentos de negocio existentes, por tanto, es fácil perder el control de las distintas APIs conforme la organización empresarial vaya creciendo. Por este motivo, surgen los sistemas API Management, cuyo objetivo principal es facilitar la administración de las APIs, centralizando en un único punto funciones de administración, seguridad, publicación, estadísticas… de las distintas APIs de una organización.

Debido al papel fundamental que desempeñan los sistemas API Management dentro del mosaico tecnológico actual es de vital importancia la protección de los datos que se exponen mediante APIs, por tanto el presente trabajo de fin de máster se centra en el estudio de los distintos servicios y mecanismos de seguridad que ofertan un conjunto de la población sometida a estudio de las grandes soluciones de API Managers actuales.

Se abordará el análisis de los mecanismos de autenticación, autorización, encriptación, funciones de seguridad y protección de ataques de los distintos sistemas API Management a estudiar, con el fin de realizar una comparativa donde se expongan las debilidades y fortalezas de cada sistema.

## 1.2. Objetivos

El objetivo principal del presente trabajo de fin de máster es analizar los mecanismos, controles y protecciones en arquitecturas basadas en API Management. Para el cumplimiento del objetivo anterior ha sido necesario trazar la siguiente serie de objetivos secundarios:

- Estudiar la arquitectura y componentes de un sistema API Management.

- Realizar un estudio relativo al marco de los sistemas API Management para extraer los sistemas con mayor impacto y uso en la actual realidad tecnológica.

- Analizar las distintas amenazas de seguridad existentes en el ámbito de las APIs REST.

- Identificar y describir los distintos mecanismos y servicios de seguridad que poseen cinco importantes soluciones API Management de la actualidad para evaluar su grado de protección frente a ataques actuales de seguridad.

- Establecer una comparativa, desde el punto de vista de la seguridad, de las soluciones API Management sometidas a estudio.

- Exponer las distintas conclusiones derivadas de la investigación y proponer distintas líneas futuras de investigación.

## 1.3. Metodología

La metodología a seguir para el desarrollo y cumplimiento de los objetivos del presente trabajo de fin de máster se divide en seis fases diferenciadas:

- Definición y establecimiento del plan de trabajo: En esta fase se incidirá en el problema a resolver, se describirán los objetivos y metodología usada y se desglosará la totalidad del trabajo en tareas que serán organizadas de manera temporal.

- Estudio de la arquitectura: Se analizará en esta fase los distintos componentes que conforman un sistema API Management y se incidirá en la funcionalidad de cada elemento perteneciente a la arquitectura.

- Examen de vulnerabilidades y ataques: En esta fase se estudiarán los distintos riesgos, desde el punto de vista de la seguridad, que sufren las APIs REST en la actualidad.

- Estudio de mercado y selección de soluciones: En esta fase se examinarán las distintas soluciones API Management existentes en el mercado actual y se seleccionarán aquellas de mayor acogida y prestaciones.

- Análisis de servicios de seguridad: Una vez seleccionados las soluciones API Management a estudiar, se analizarán en esta fase los servicios de seguridad que ofertan dichas soluciones.

- Síntesis y comparativa: Finalmente en esta fase se realizará una comparativa a modo de resumen donde se reflejarán las debilidades y fortalezas de cada una de las soluciones.

## 1.4. Listado de tareas

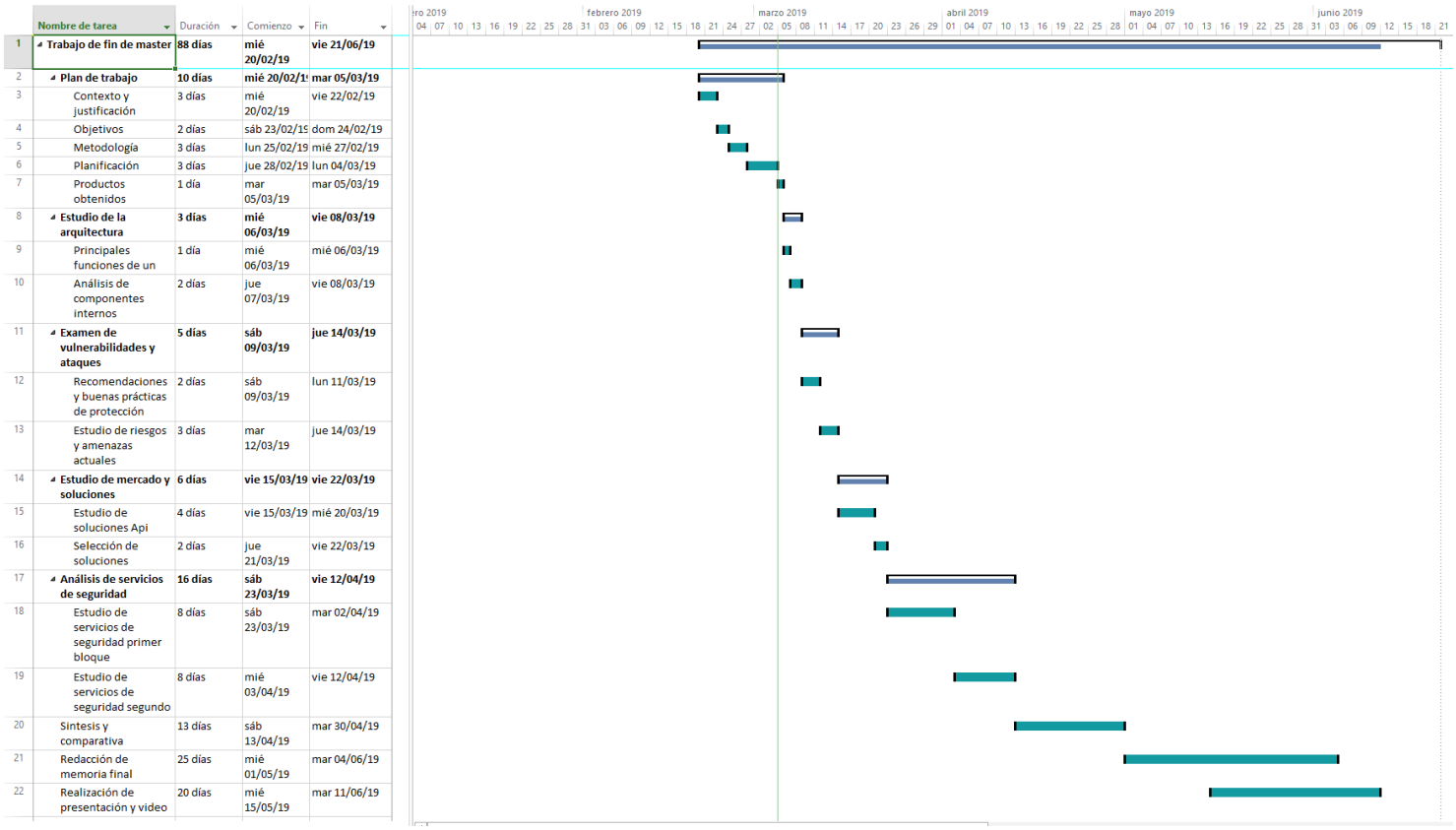
A continuación se detalla el desglose de las principales tareas identificadas en el presente plan de trabajo:

- Plan de trabajo:
    - Contexto y justificación.
    - Objetivos.
    - Metodología.
    - Planificación.
    - Productos obtenidos.
- Estudio de la arquitectura:
    - Principales funciones de un sistema API Management.
    - Análisis de componentes internos.
- Examen de vulnerabilidades y ataques:
    - Recomendaciones y buenas prácticas de protección.
    - Estudio de riesgos y amenazas actuales.
- Estudio de mercado y selección de soluciones:
    - Estudio de soluciones API Management consolidadas.
    - Selección de soluciones.

- Análisis de servicios de seguridad:
    - Estudio de los servicios de seguridad del primer bloque de las soluciones API Management seleccionadas.
    - Estudio de los servicios de seguridad del segundo bloque de las soluciones API Management seleccionadas.
- Síntesis y comparativa.
- Redacción de memoria final.
- Realización del video y presentación.

## 1.5. Planificación del trabajo

En la *Figura 1. Planificación temporal* se define la siguiente planificación temporal para el presente trabajo de final de máster:



| | Nombre de tarea | Duración | Comienzo | Fin |
|---|---|---|---|---|
| 1 | **Trabajo de fin de master** | **88 días** | **mié 20/02/19** | **vie 21/06/19** |
| 2 | **Plan de trabajo** | **10 días** | **mié 20/02/19** | **mar 05/03/19** |
| 3 | Contexto y justificación | 3 días | mié 20/02/19 | vie 22/02/19 |
| 4 | Objetivos | 2 días | sáb 23/02/19 | dom 24/02/19 |
| 5 | Metodología | 3 días | lun 25/02/19 | mié 27/02/19 |
| 6 | Planificación | 3 días | jue 28/02/19 | lun 04/03/19 |
| 7 | Productos obtenidos | 1 día | mar 05/03/19 | mar 05/03/19 |
| 8 | **Estudio de la arquitectura** | **3 días** | **mié 06/03/19** | **vie 08/03/19** |
| 9 | Principales funciones de un | 1 día | mié 06/03/19 | mié 06/03/19 |
| 10 | Análisis de componentes internos | 2 días | jue 07/03/19 | vie 08/03/19 |
| 11 | **Examen de vulnerabilidades y ataques** | **5 días** | **sáb 09/03/19** | **jue 14/03/19** |
| 12 | Recomendaciones y buenas prácticas de protección | 2 días | sáb 09/03/19 | lun 11/03/19 |
| 13 | Estudio de riesgos y amenazas actuales | 3 días | mar 12/03/19 | jue 14/03/19 |
| 14 | **Estudio de mercado y soluciones** | **6 días** | **vie 15/03/19** | **vie 22/03/19** |
| 15 | Estudio de soluciones Api | 4 días | vie 15/03/19 | mié 20/03/19 |
| 16 | Selección de soluciones | 2 días | jue 21/03/19 | vie 22/03/19 |
| 17 | **Análisis de servicios de seguridad** | **16 días** | **sáb 23/03/19** | **vie 12/04/19** |
| 18 | Estudio de servicios de seguridad primer bloque | 8 días | sáb 23/03/19 | mar 02/04/19 |
| 19 | Estudio de servicios de seguridad segundo | 8 días | mié 03/04/19 | vie 12/04/19 |
| 20 | Síntesis y comparativa | 13 días | sáb 13/04/19 | mar 30/04/19 |
| 21 | Redacción de memoria final | 25 días | mié 01/05/19 | mar 04/06/19 |
| 22 | Realización de presentación y video | 20 días | mié 15/05/19 | mar 11/06/19 |

**Figura 1. Planificación temporal**

## 1.6. Productos obtenidos

En el presente trabajo de fin de máster se han definido los siguientes productos o entregables:

| Nombre Entregable | Tareas |
|---|---|
| **PEC1** | • Plan de trabajo |
| **PEC2** | • Estudio de la arquitectura<br>• Examen de vulnerabilidades y ataques<br>• Estudio de mercado y soluciones primer bloque |
| **PEC3** | • Estudio de mercado y soluciones segundo bloque<br>• Síntesis y comparativa |
| **PEC4** | • Memoria final de trabajo de fin de máster |
| **PEC5** | • Vídeo/presentación |

**Tabla 1. Productos obtenidos**

# 2. Estudio de la arquitectura API Management

## 2.1. Principales funciones de un API Management

Debido al creciente número de APIs de distinta tipología que conviven en la realidad tecnológica actual, los sistemas API Manager están aumentando su popularidad ya que estos sistemas son capaces de administrar y gestionar de manera centralizada distintas tareas y funciones enmarcadas dentro del ciclo de vida de las APIs. En la *Figura 2. Crecimiento APIs Web* se representa el crecimiento que han experimentado las APIs Web en el periodo comprendido entre el año 2005 y el año 2018:

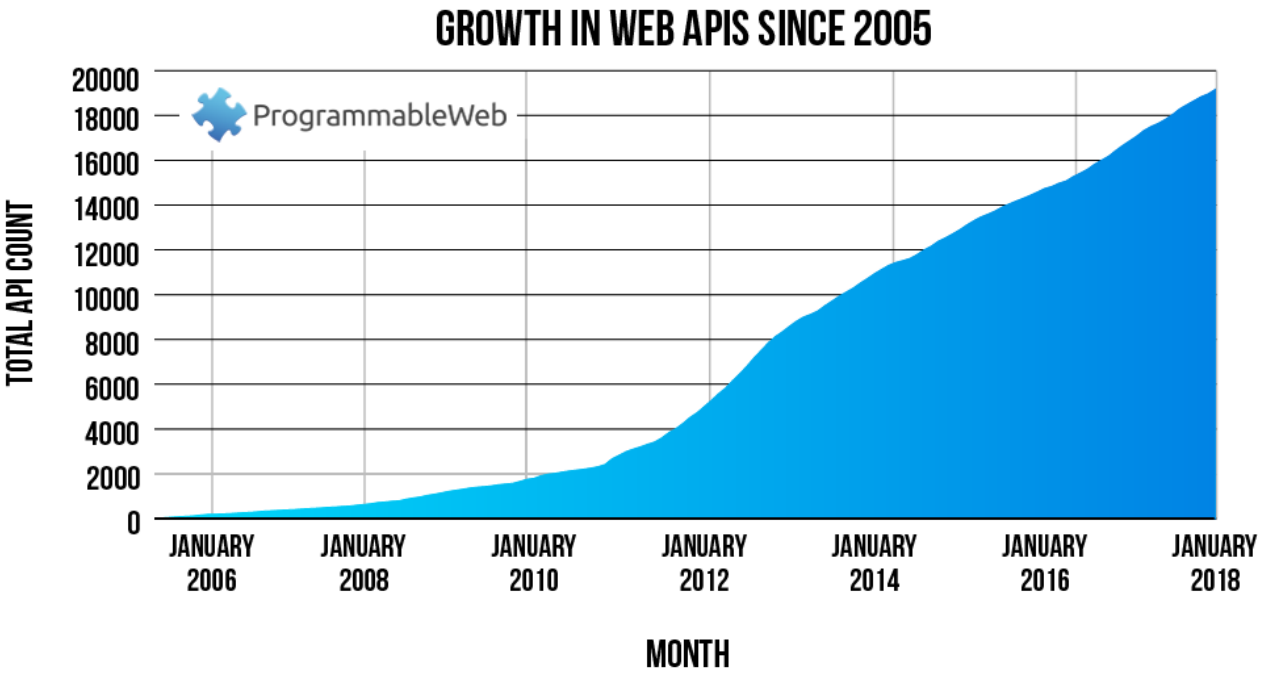



**Figura 2. Crecimiento APIs Web**

**Fuente:https://www.programmableweb.com/news/research-shows-interest-providing-apis-still-high**

Debido a este gran crecimiento de las APIs en los últimos años, han surgido problemas en el desarrollo y mantenimiento de las APIs. Estos problemas involucran tanto a los propietarios de las APIs como a los propios consumidores, siendo algunos de los problemas más relevantes:

- Securización.
- Documentación.
- Escalabilidad.
- Versionado y evolución.
- Gobierno del ciclo de desarrollo.

Como consecuencia de la existencia de los diversos problemas anteriores surgen los sistemas API Management para aportar soluciones y satisfacer las necesidades de los propietarios y consumidores de las APIs.

Los sistemas API Management son sistemas que agrupan distintas procesos de actuación directa sobre las APIs Web. Un sistema de API Management es el encargado de la realización de procesos [1] [2] de:

- **Publicación**: Publicar APIs haciéndolas accesibles a los distintos consumidores interesados, en el ámbito de consumo correspondiente (publicación en Internet o privada).

- **Escalabilidad**: Asegurar la escalabilidad de las distintas APIs, realizando funciones de balanceado u otras técnicas que aseguren el nivel de disponibilidad deseado para cada API.

- **Gobierno**: Auditar y supervisar las APIs, focalizando este control en aspectos como:

    - Monitorización de recursos.

    - Tarificación.

    - Generación y explotación de métricas funcionales y de sistema.

- **Seguridad**: Garantizar la seguridad de las APIs, usuarios y recursos alojados en el sistema de API Management.

- **Empaquetado**: Posibilidad de empaquetar un mismo servicio atendiendo a distintos criterios, ya que pueden existir distintos clientes interesados en aspectos y funcionalidades distintas de una misma API.

- **Versionado**: Proveer un mecanismo fácil que posibilite la evolución de la API en sus distintas versiones y que asegure la compatibilidad con versiones anteriores.

- **Staging**: Ofertar herramientas que permitan desplegar distintas versiones de las APIs en distintos entornos de ejecución para poder cumplir con el ciclo de vida del software.

## 2.2. Análisis de componentes API Management

Los sistemas API Management están formados por una serie de componentes [3] [4] con diversas funciones que componen su arquitectura. En la *Figura 3. Componentes sistemas API Management* se presentan los componentes o elementos principales de los sistemas API Management:

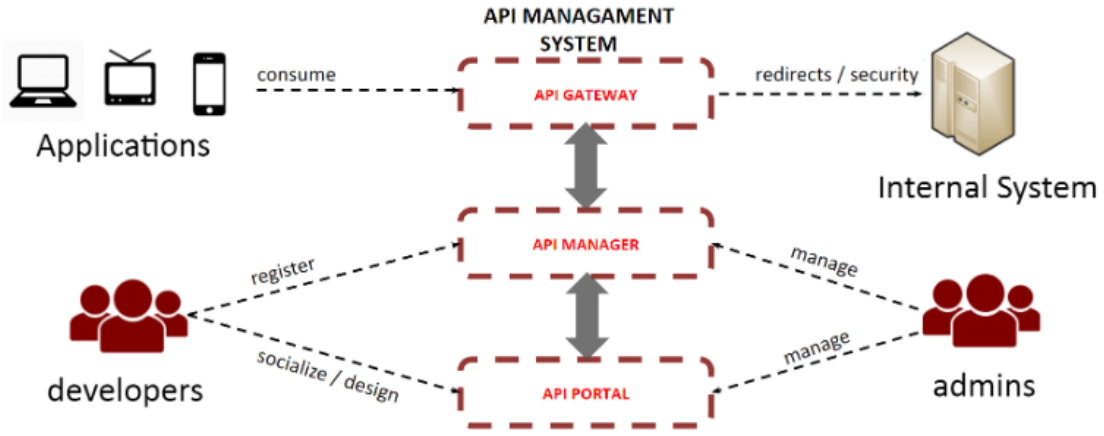



**Figura 3. Componentes sistemas API Management**

**Fuente: https://www.paradigmadigital.com/dev/api-management-que-es-y-para-que-sirve/**

- **API Manager**: La principal misión de este componente es la de proveer mecanismos de configuración y publicación de las APIs. De entre sus funciones destacan:

    - Publicación: Publica las APIs que se alojan en el componente y definen un endpoint mediante el cual los distintos consumidores podrán acceder al API.

    - Gestión del API life-cycle: Permite gestionar las distintas etapas pertenecientes al ciclo de vida de la API, posibilitando la modificación de aspectos como la versión de la API o la retirada de la misma si en un instante temporal se considerara deprecada.

    - Edición: El componente API Manager permite modificar distintos aspectos de la API en caliente, como la edición de la propia interfaz que expone el API a los consumidores, el endpoint de invocación, políticas de seguridad o la propia desactivación de la API.

    - Monitorización del uso: Generación de estadísticas (SLAs, QoS…) y creación de alertas relacionadas con las distintas métricas de consumo de recursos del sistema y de las APIs a monitorizar.

    - Gestión de políticas de seguridad: El componente API Manager es el encargado de administrar y configurar determinados aspectos de seguridad vinculados con las APIs como mecanismos de blacklist, control de tráfico…

- **API Gateway**: Es el componente más importante de la arquitectura de los sistemas API Management, de hecho, los primeros sistemas API Management poseían únicamente este componente. Componente cuyo principal objetivo es comunicar, mediante APIs, los distintos servicios

con sus posibles consumidores. Las funciones más relevantes del componente API Gateway son las siguientes:

- Enrutamiento: Proporcionar el correcto enrutado de los mensajes en función del endpoint de invocación asociado a la API o del propio contenido del mensaje enviado en la comunicación. Esta función es una de las más significativas dentro de los sistemas API Management, puesto que es la característica que posibilita la comunicación usuario-api-servicio.

- Multi protocolo: Soporte a distintos protocolos a la hora de publicar las APIs, por ejemplo publicación del API mediante HTTP o HTTPS. A su vez, el componente API Gateway permite la coexistencia de múltiples protocolos con los que la API puede interaccionar como: jdbc, pop3/imap, smtp, ftp...

- Multi formato: El API Gateway es capaz de transformar los mensajes de recepción/destino de la API, como por ejemplo realizar una transformación XML a JSON o viceversa.

- Monitorización de tráfico: Inspección del tráfico de entrada y salida del API Management, llegando a realizar filtrados del tráfico (por protocolo, dirección IP…) en caso de ser convenientes.

- Gestión de la seguridad del API: El componente API Gateway permite dotar a las APIs de distintos aspectos relacionados con la seguridad como:

  - Mecanismos de autenticación.

  - Mecanismos de autorización.

  - Cifrado del tráfico.

- **API Portal**: Componente que expone una interfaz gráfica a los administradores del API Management (esta interfaz gráfica a menudo es web) para que puedan configurar y administrar de manera visual características relativas a todos los componentes de la arquitectura del sistema API Management. Existen dos tipos de portales claramente diferenciados:

  - Portal de administración: Enfocado principalmente a los administradores del sistema API Management. Este portal cuenta con las siguientes características:

    - Listado de APIs publicadas con posibilidad de edición de su interfaz.

    - Almacén de certificados de seguridad.

- Configuración de accesos a bases de datos.
- Opciones de securización a nivel de API.
- Monitorización de recursos del sistema: Consumo de CPU, memoria, último reinicio del sistema…

- Portal de estadísticas: Este portal está dirigido a los responsables de los servicios, de esta manera, el administrador del sistema API Management puede facilitar credenciales a los responsables de cada servicio, que es invocado mediante un API, para que puedan visualizar y definir métricas relativas al consumo de su servicio, como por ejemplo cuántas veces se ha invocado en el último mes a su servicio, o la tasa media de llamadas erróneas en la última hora. Mediante este portal, los responsables de los servicios son capaces de explotar la información de consumo expuesta en diferentes dashboards del portal, para obtener un mayor control o hacer crecer su negocio.

# 3. Examen de vulnerabilidades y ataques

En este capítulo se realizará un recorrido por las distintas vulnerabilidades y ataques que en la actualidad hacen que la seguridad de las APIs, y por tanto de los sistemas API Management, quede comprometida. Posteriormente se incidirá en una serie de recomendaciones y buenas prácticas para incrementar el nivel de seguridad de las APIs y fortalecer el nivel de protección para así disminuir el nivel de exposición a diferentes ataques externos.

## 3.1. Estudio de riesgos y amenazas actuales

La naturaleza web intrínseca de los sistemas API Management hace que estos sistemas compartan las vulnerabilidades de las aplicaciones web. De acuerdo con OWASP (Open Web Application Security Project), que es un proyecto de código abierto dedicado a identificar y luchar contra las vulnerabilidades que hacen que el software sea inseguro, los principales diez riesgos [5] [6] y vulnerabilidades de las aplicaciones web son:

- **Inyección**: Los ataques de inyección pueden ser de diferentes tipos: SQL, No SQL, LDAP… Sin embargo, todos los ataques de inyección tienen como denominador común el aprovechamiento de una fuente de datos para provocar un comportamiento no deseado o malicioso en un intérprete. De esta manera, un usuario mal intencionado podría inyectar código en un servicio web para conseguir por ejemplo, las credenciales de los usuarios del sistema.

- **Pérdida de autenticación**: Los mecanismos de autenticación mal implementados o vulnerables, pueden ser utilizados por un usuario malintencionado para la suplantación de identidades, secuestros de sesión, robo de credenciales…

- **Exposición de datos sensibles**: Una gran mayoría de aplicaciones web no protegen de manera adecuada los datos sensibles con los que la propia aplicación interacciona. Por tanto, un atacante explotando una vulnerabilidad del sistema podría robar estos datos personales, para llevar acabos delitos como suplantación de identidad, fraudes con tarjetas de crédito u otro tipo de delitos.

- **Entidades externas XML**: Los atacantes explotan la capacidad de muchos procesadores XML que permiten especificar una entidad externa en base a una URL, de esta forma, un atacante puede insertar esta entidad externa en un documento XML y explotar el código vulnerable del procesador XML usado en la aplicación. Mediante ataques de este tipo un atacante puede realizar ataques de denegación de servicio, extracción de datos, realizar una solicitud remota, análisis de puertos activos y ejecutar código malicioso.

- **Pérdida de control de acceso**: El establecimiento de una adecuada política de autorización es crucial para la correcta securización del sistema. Es imprescindible que el sistema cuente con un mecanismo granular de autorización que permita el establecimiento de los permisos únicamente necesarios a los usuarios sobre los distintos recursos del sistema. Si el sistema no posee un adecuado mecanismo de control de acceso, un atacante podría aprovechar esta deficiencia para acceder a recursos no permitidos, modificar permisos de otros usuarios o incluso editar o borrar ciertos recursos alojados en el sistema.

- **Configuración de seguridad inadecuada**: A menudo, por falta de conocimiento o por olvido, los administradores de los sistemas descuidan determinados ámbitos de configuración de la seguridad de sus sistemas. Ejemplos de configuraciones incorrectas de los sistemas pueden ser la falta de actualizaciones de los componentes del sistema, el uso de cabeceras HTTP erróneas o mensajes de error en el que se muestra información sensible o no deseada.

- **Cross-site scripting (XSS):** Esta vulnerabilidad permite la inyección de código para lograr la ejecución de scripts maliciosos. Mediante la explotación de esta vulnerabilidad un atacante puede realizar un secuestro de sesión, modificar los elementos de un sitio web (defacement) o incluso redireccionar a un usuario a un sitio web malicioso.

- **Deserialización insegura:** Cuando una aplicación recibe los objetos serializados de los usuarios si no se ha realizado un adecuado control del proceso de deserialización puede ocasionar graves inconvenientes en la seguridad. Un usuario malintencionado podría mandar objetos maliciosos serializados a la aplicación, de tal forma, que cuando la aplicación en recepción, realice el proceso de deserialización cause un comportamiento anómalo en el sistema. Con el uso de esta técnica un atacante podría ejecutar código de manera remota en el servidor y realizar ataques de inyección, de repetición y de elevación de permisos.

- **Componentes de la aplicación vulnerables**: En una aplicación web existen multitud de componentes que interactúan entre sí. Estos componentes poseen los mismos permisos de ejecución que la aplicación, por lo que si alguno de los componentes queda expuesto, la seguridad de la aplicación se verá comprometida y el atacante se podría hacer con el control del sistema objetivo.

- **Monitorización insuficiente**: OWASP expone que el tiempo medio de detección de una brecha en la seguridad de un sistema de una organización es superior a dos cientos días. Lo anteriormente expuesto se justifica en que la mayoría de los responsables de los sistemas no cuenta con un adecuado sistema de monitorización, o no realizan ningún procesado de los logs de los distintos componentes, de tal forma, que en caso de detectar un comportamiento anómalo, este se notifique mediante el lanzamiento de algún tipo de alerta.

## 3.2. Recomendaciones y buenas prácticas de seguridad

En este apartado se presentan una serie de recomendaciones planteadas por OWASP [7] para proteger los sistemas de API Management y las aplicaciones web, en base a los riesgos y vulnerabilidades expuestas en el apartado anterior:

| Vulnerabilidad | Medidas de protección |
|---|---|
| **Inyección** | • Uso de herramientas de mapeo relacional de objetos (ORMs).<br>• Validación de los datos introducidos por el usuario que procesará la aplicación.<br>• Escapado de caracteres en la realización de consultas dinámicas.<br>• Uso de funciones SQL para controlar la devolución máxima de registros, como por ejemplo *LIMIT*. |
| **Pérdida de autenticación** | • Uso de autenticación multi-factor para evitar ataques automatizados.<br>• No usar credenciales por defecto, implementar controles para evitar usar contraseñas débiles e implementar una política de caducidad de contraseñas.<br>• Registrar los fallos de autenticación e implementar mecanismos de bloqueo de login temporal ante inicios de sesión fallidos.<br>• Utilizar un gestor de sesión en la aplicación. |
| **Exposición de datos sensibles** | • No almacenar datos sensibles de manera innecesaria.<br>• Cifrado de datos sensibles en almacenamiento y en tránsito.<br>• No guardar en caché datos sensibles.<br>• Almacenamiento de credenciales mediante el uso de funciones hash robustas. |
| **Entidades externas XML** | • Usar estándares de datos menos complejos que XML, como por ejemplo JSON.<br>• Desactivación del procesado de entidades externas XML en los parseadores de la aplicación.<br>• Validación por esquema (XSD) del XML recibido por la aplicación.<br>• Uso de firewalls de tipo aplicaciones web (WAF). |
| **Pérdida de control de acceso** | • Implementación de política de denegación por defecto en el acceso a los recursos.<br>• Deshabilitar el listado de directorios del servidor web.<br>• Registrar errores de control de acceso e implementar alertas ante eventos no deseados.<br>• Limitación de la tasa de acceso a la API para evitar los daños de ataques automatizados. |
| **Configuración de seguridad inadecuada** | • No instalar componentes innecesarios en la aplicación.<br>• Implementar una política de actualizaciones y de gestión |

| | |
|---|---|
| | de parches.<br>• Envío de cabeceras de seguridad a los clientes.<br>• Segmentar la arquitectura de los componentes de la aplicación para que exista una separación efectiva y segura. |
| **Cross-site scripting** | • Uso de frameworks que cuenten con mecanismos efectivos contra ataques de tipo Cross-Site Scripting.<br>• Habilitar una política de seguridad de contenido (CSP). |
| **Deserialización insegura** | • No aceptar objetos serializados de fuentes no confiables.<br>• Realización de verificaciones en el proceso de deserialización del objeto recibido en la aplicación.<br>• Aislar el código responsable de la deserialización y dotarlo de privilegios mínimos.<br>• Registrar los fallos de deserialización y crear alertas que notifiquen fallos y procesos de deserialización frecuentes asociados a un mismo usuario. |
| **Componentes de la aplicación vulnerables:** | • Eliminar componentes y/o dependencias que no se encuentren en uso en la aplicación.<br>• Seguimiento de vulnerabilidades de los componentes de la aplicación en bases de datos de vulnerabilidades.<br>• Usar componentes de confianza u oficiales. |
| **Monitorización insuficiente** | • Implementar una política de tratamiento de logs donde se especifiquen el tiempo de vida de los mismos y los eventos a monitorizar.<br>• Trazabilidad y registro de las transacciones de más relevancia dentro de la aplicación.<br>• Establecer un sistema de alertas, de tal manera, que ante cualquier actividad sospechosa, seamos notificados con prontitud.<br>• Definir un plan de respuesta ante la activación de las distintas alertas configuradas. |

**Tabla 2. Recomendaciones OWASP TOP 10 vulnerabilidades**

En adición a las distintas recomendaciones anteriores, existen otra serie de ámbitos clave en los que se debe incidir para incrementar la seguridad de las APIs:

- **HTTPS**: El endpoint mediante el cual se expone una API debe usar siempre el protocolo HTTPS, de esta forma, se protege la información en tránsito mediante cifrado y la información sensible no queda expuesta. En adición, para los servicios críticos, se debe usar el esquema de autenticación mutua del protocolo HTTPS.

- **Control de acceso**: Cada API tiene que contar con su propia configuración de los mecanismos de autenticación, autorización y gestión de sesión. A su vez, es importante contar con la presencia de un gestor de identidades, donde la autenticación se centralice en un único punto, y sea este componente el que expida las identidades.

- **API keys**: Estas claves de API son usadas para controlar el acceso a las APIs y de esta manera gobernar a los usuarios que las invocan. Con estas claves se puede reducir el impacto de los ataques de denegación de servicio, aunque se ha de tener en cuenta, que si el usuario al que se le facilita la clave para poder consumir el API compromete la clave, entonces la funcionalidad aportada por el API Key deja de tener sentido.

- **Restricción de métodos HTTP**: Es vital realizar un control de los distintos métodos HTTP necesarios para el uso de los distintos recursos de la API. La forma correcta de proceder es bloquear todos los métodos HTTP que no se usen en la API.

- **Validación de los tipos de contenido**: Definir y documentar los tipos de contenido aceptados por nuestra aplicación, y en caso de que la aplicación reciba un tipo de contenido no soportado rechazar la petición.

- **Endpoints de administración**: En lo que respecta los endpoints de las APIs es importante cumplir con las siguientes consideraciones:

    - No exponer los endpoints de administración del sistema en Internet. En caso de tener que exponer el endpoint en Internet, implementar mecanismos de autenticación robustos.

    - Restringir el acceso a los endpoints de administración mediante reglas del firewall o mecanismos basados en listas de control de acceso.

- **Información sensible en peticiones HTTP**: Es vital proteger la información sensible en las comunicaciones con objeto de protegernos ante el robo de información personal o de las contraseñas que viajan en las comunicaciones consumidores-API-servicios. Para prevenir que la información sensible sea comprometida se debe cumplir:

    - En las peticiones HTTP en las cuales se use los métodos POST y PUT y se envíe información sensible, esta información debe ser incluida en el cuerpo de la petición o en las cabeceras HTTP.

    - En las peticiones HTTP en las que se use el método GET y se desee enviar información sensible, esta información irá contenida en las cabeceras HTTP.

- **Códigos de respuesta HTTP**: El protocolo HTTP define un código de estado que poseerá un valor u otro en función de la casuística de la comunicación que se produzca. La correcta asignación del código de estado en las distintas respuestas que una aplicación devuelva juega un papel fundamental desde el punto de vista de la seguridad, ya que estos códigos nos permiten detectar situaciones anómalas y posibles ataques.

# 4. Estudio de mercado y selección de soluciones

## 4.1. Análisis de mercado API Management

El auge de la movilidad tecnológica, la proliferación de las aplicaciones móviles, el aumento del número de redes sociales, el creciente nacimiento y demanda de APIs de índole privada y pública, el gran número de APIs Web y la emergencia de paradigmas como IoT o Big Data, han sido factores claves que han convertido al mercado de los sistemas API Management en un mercado fuertemente emergente.

Según importantes empresas de investigación de mercados [8] [9] como *Markets And Markets* y *Market Research Future* se espera que el mercado de los sistemas API Management goce de una tasa de crecimiento anual compuesta del 32.9%, empezando el periodo de estudio en el año 2018 y finalizando en el año 2023. Este porcentaje de la tasa de crecimiento se traduce en un aumento de 1.2 billones de dólares americanos (en el año 2018) a 5.1 billones de dólares americanos (en el año 2023).

El estudio de la empresa *Markets And Markets* y *Market Research* titulado "*API Management Market by Solution (API Platform, API Analytics, API Security), Service (Integration and Implementation, Consulting, Support and Maintenance, Training), Deployment Type, Organization Size, Industry, and Region - Global Forecast to 2023*" indica que la mayor parte del mercado de API Management se concentrará [10] en la regiones de Norte América y Canadá. Estas regiones han adoptado fuertemente tecnologías como Cloud Computing, IoT, movilidad, Big Data, analíticas en tiempo real…, por tanto, en dichas regiones está previsto que se concentre la mayor parte del sector del mercado de API Management. A la región de Norte América le siguen en orden de importancia las regiones de Europa, Asía Pacífica, América Latina, Oriente Medio y África. En la *Figura 4. Crecimiento API Management* se muestra la evolución del crecimiento esperado en el mercado de los sistemas API Management en función de la región geográfica:

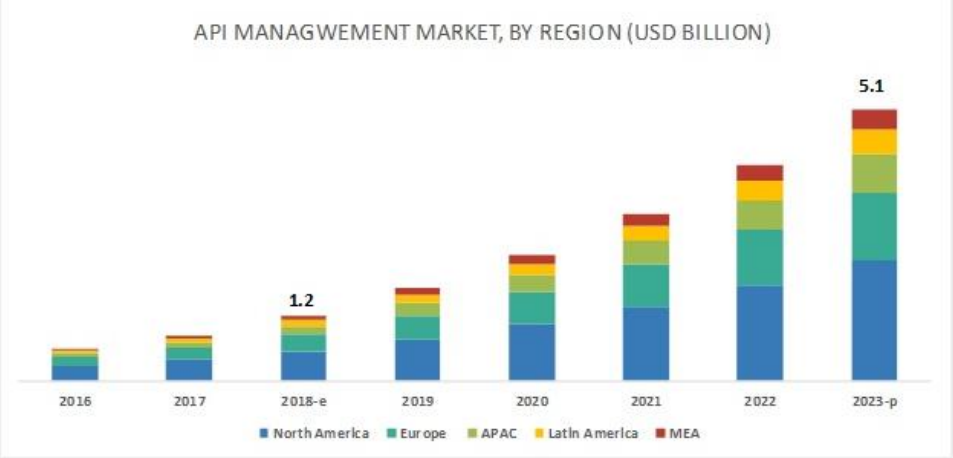



**Figura 4. Crecimiento API Management**

**Fuente: https://www.marketsandmarkets.com**

Dado el emergente escenario actual del mercado de los sistemas API Management existen distintos fabricantes tecnológicos que se han consolidado o están apostando en el desarrollo de soluciones API Management. En el informe "*Magic Quadrant for Full Life Cycle API Management"* elaborado por la prestigiosa consultora *Gartner* [11], se presenta un diagrama bautizado como *Magic Quadrant* donde se refleja el estado y relevancia de los distintos fabricantes de soluciones API Management, véase *Figura 5. Magic Quadrant Gartner*:


**Figura 5. Magic Quadrant Gartner**

**Fuente: https://www.gartner.com**

El diagrama anterior realiza una evaluación de los distintos fabricantes en función de su capacidad de ejecución y su visión estratégica, ofertando a simple vista una visión rápida del crecimiento y posición de mercado en el que se encuentra cada fabricante.

Con objeto de cubrir distintos estados de madurez y ampliar la riqueza del análisis de las soluciones cubiertas, analizando soluciones API Management de fabricantes que ocupan distintas posiciones de mercado, en el presente trabajo de fin de máster se estudiarán dos de los fabricantes líderes del sector y un fabricante perteneciente al resto de cuadrantes del *Magic Quadrant*, optando

siempre por soluciones consolidadas en el mercado, que no se encuentran en fase de testeo y que en consecuencia, están disponibles como producto final. Por tanto, se estudiarán los aspectos de seguridad ofertados por los siguientes sistemas API Management:

- Líderes:
    - **Apigee** (*Google*).
    - **CA API Management** (*CA Technologies*).
- Aspirantes: En esta categoría se analizará la solución **Amazon API Gateway** perteneciente a la suite *Amazon AWS*.
- Fabricantes nicho: Se analizará **Microsoft's Azure API Management** del fabricante perteneciente a la suite *Azure* de *Microsoft*.
- Visionarios: En esta agrupación de fabricantes en los que se tienen grandes expectativas se examinará la solución **WSO2 API Manager** de la suite de productos WSO2.

## 4.2. Selección de soluciones

En este apartado se presentarán y describirán brevemente las distintas soluciones API Management seleccionadas para su estudio en el presente trabajo de fin de máster.

### 4.2.1. Apigee

La empresa *Apigee* fue comprada por Google en septiembre de 2016. *Apigee* posibilita la administración y control de las distintas etapas involucradas en el ciclo de vida de la API. La arquitectura de *Apigee* está compuesta de diversos componentes siendo el componente principal de la arquitectura *Apigee Edge*. El componente *Apigee Edge* tiene como misión la administración global de la plataforma, y su instalación es posible tanto en entorno cloud como en cualquier servidor privado.

*Apigee Sense*, es el componente encargado de gestionar la capa de seguridad de la arquitectura. A su vez, otro de las características a destacar de la arquitectura de *Apigee* es el componente de monetización, que permite a las empresas tarificar a los usuarios en función del uso que hagan de sus APIs.

La funcionalidad de *Apigee* está disponible en diversos planes de servicio, ofertando *Google* desde un plan de prueba gratuito hasta planes de pago para grandes corporaciones.

El sistema API Management *Apigee* posee las siguientes características [12] principales:

- Transformación de protocolos: Convertir de un protocolo a otro, como SOAP, REST…

- Empaquetado personalizado de API: Permite asignar a cada api un empaquetado ad-hoc con diferentes precios y umbrales de uso, en función, de la necesidad de cada cliente.

- Control del tráfico: Posibilidad de asignación de políticas en función de cuotas, límites de frecuencia y retención de picos.

- Análisis y monitorización: Filtrado del rendimiento del sistema, trazas de eventos anómalos y estadísticas de uso en función del usuario, aplicación o API.

- Filtrado de seguridad: Existe la posibilidad de realizar filtrados por dirección IP y de realizar validaciones en los formatos de los mensajes entrantes XML y JSON, también *Apigee* cuenta con un mecanismo para detectar bots maliciosos.

- Monetización: Con *Apigee* es posible aplicar planes de tarificación flexibles con la posibilidad de incluir precios dinámicos, tarificación internacional y mecanismos de facturación en base al uso de la API.

- Administración global: Aplicación de políticas de las APIs de administración y seguridad, a nivel global en caso de ser necesario.

- Pasarelas federadas: Posibilidad de desplegar y ejecutar las APIs en los entornos donde se encuentren las aplicaciones y de administrarlas de manera centralizada y remota.

- Portal para desarrolladores: Portal que facilita la administración de desarrolladores, APIs y su documentación asociada y versionados.

Finalmente en la *Figura 6. Servicios Apigee* se ilustra como *Apigee* es capaz de proporcionar algunos de los servicios anteriormente citados a los distintos agentes implicados en el escenario del ciclo de vida de las APIs:

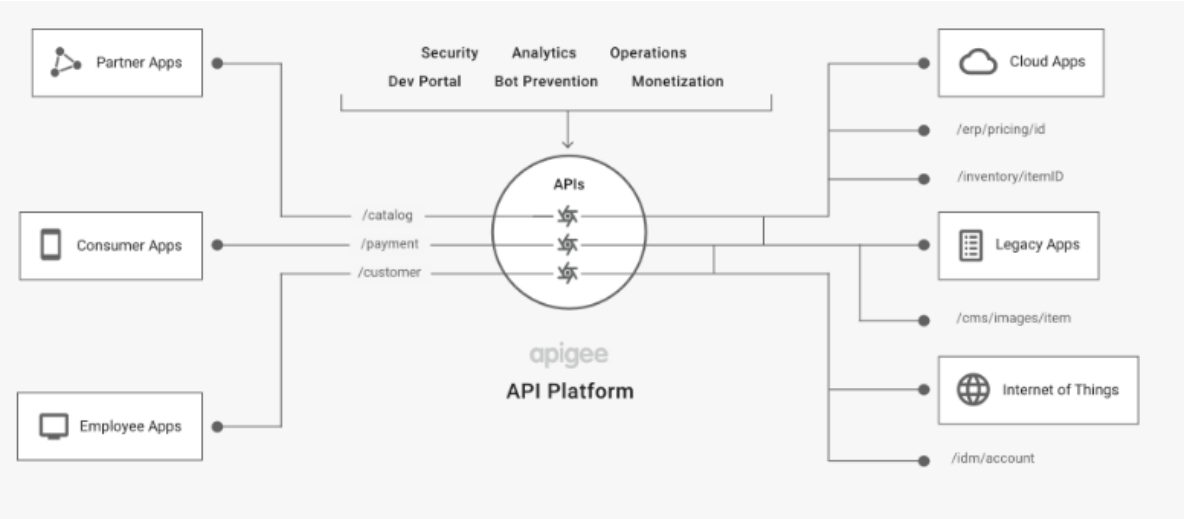



**Figura 6. Servicios Apigee**

**Fuente: https://cloud.google.com/apigee-api-management**

### 4.2.2. CA API Management

*CA API Management* es el principal producto de la compañía *CA Technologies*, este producto aglutina diversos componentes para ofrecer una amplia funcionalidad con el fin de simplificar y unificar la administración de las APIs. *CA API Management* se divide en los siguientes componentes:

- **CA API Gateway**: Componente [13] encargado de la securización y el gobierno de las APIs, el componente cuenta con las siguientes características:

    - Rendimiento: Facilidad de escalado y posibilidad de realizar despliegues en alta disponibilidad con el fin de garantizar que las APIs que se consideren críticas siempre estén disponibles.

    - Facilidad de uso: El componente *API Gateway* oferta una vista centralizada y en tiempo real del estado de las APIs. En adición, este componente permite la migración de las APIs entre los distintos entornos de ejecución (desarrollo, pruebas y producción) mediante herramientas de administración globales.

    - Extensión de la funcionalidad: La funcionalidad de la totalidad del sistema es plenamente extensible mediante el framework de plug-ins que proporciona *CA*.

    - Mecanismos de seguridad: Permite la securización de los distintos recursos corporativos mediante un sistema de control centralizado. API Gateway provee integración con la mayoría de los sistemas de gestión de acceso e identidades del mercado.

- Control de tráfico: Priorización del tráfico con el fin de garantizar que las APIs permanezcan disponibles a lo largo del tiempo.

- Transformación y orquestación: Capacidad para conectar distintos servicios, con diferentes protocolos y formatos de datos, entre sí.

- Flexibilidad: Soporte a diferentes modelos de despliegue, siendo compatible con diversas plataformas como *Docker*, *AWS* o *Azure*.

- **CA API Management SaaS**: Este componente [14] es una plataforma de administración en la nube, aunque CA también oferta un enfoque de instalación híbrido entre la nube y el servidor privado de la organización. Engloba la mayor parte de la funcionalidad del componente *CA API Gateway*, pero con la ventaja de que el cliente no tiene que preocuparse instalar el componente de administración, ya que este componente estará disponible en la nube.

- **CA Live API Creator**: Componente [15] que permite la creación de APIs y de microservicios. De entre su funcionalidad destaca:

  - Creación eficiente y personalizable: *CA Live API Creator* persigue la creación rápida de APIs y microservicios mediante una interfaz visual. Además esta interfaz gráfica de creación posee una gran variedad de opciones para personalizar distintos ámbitos en el proceso de creación de la API o del microservicio.

  - Lógica reactiva: Permite modificar configuración de la API o del microservicio ante el suceso de determinados eventos configurables.

  - Soporte de distintos brokers de mensajería: Compatible con publicadores y listeners de mensajería como *MQTT*, *Kafka*, *JMS* o *RabbitMQ*.

  - Integración de datos: *CA Live API Creator* posibilita la integración con diversas fuentes de datos como bases de datos SQL, No-SQL y servicios de datos cloud.

- **CA API Developer Portal**: Componente [16] que comprende la funcionalidad de publicación y descubrimiento de las APIs. Posee las siguientes funcionalidades:

  - Facilidad de publicación: El componente brinda la capacidad de publicar APIs de manera rápida y simple, ofertando capacidad de autogeneración de documentación del API.

  - Catálogo de APIs: Acceso rápido al catálogo de todas las APIs, donde se puede consultar el código o la documentación asociada a cada una de ellas.

- Estadísticas de APIs: Generación de informes y dashboards que ayudan al cliente a obtener una visión rápida de los KPIs y distintas métricas importantes para su organización.

- Flexible despliegue: El API Portal puede ser desplegado en distintas plataformas como *Docker* o soluciones cloud.

- Integración DevOps para las APIs: *CA API Developer Portal* provee mecanismos de integración continua para gestionar el ciclo de vida de las APIs de manera más cómoda y rápida.

- **CA Mobile API Gateway**: Este componente [17] pone el foco en el ámbito de la movilidad, ofreciendo una solución API Gateway para las APIs móviles. Destacan del componente las siguientes características:

  - Seguridad: El componente es compatible con los principales sistemas de administración de identidad y control de acceso móviles.

  - Facilidad de desarrollo: Exposición de un conjunto de herramientas basadas en estándares para incrementar la agilidad en el desarrollo de las APIs móviles.

  - Soporte a la visión IoT mediante la provisión de un sistema escalable capaz de albergar un gran número de APIs.

- **CA Microgateway**: Componente [18] con funcionalidad similar al componente CA API Gateway pero orientado arquitecturas de microservicios.

### 4.2.3. Amazon API Gateway

Amazon lanzó en el año 2015 el producto *Amazon API Gateway* dentro de su suite de productos *Amazon Web Services* (AWS). *Amazon Api Gateway* permite a los desarrolladores publicar, mantener, monitorizar y securizar APIs. Una de las principales diferencias de *Amazon API Gateway* respecto a sus principales competidores es que un producto puramente cloud en el que no se paga una cuota mensual por los servidores cloud, sino por el uso que se haga de los recursos (estrategia server-less). *Amazon* oferta un plan gratis durante un año y luego cuenta con varios planes en su catálogo que poseen una tarificación distinta en base a diferentes parámetros de uso como el número de llamadas que reciban las APIs o los minutos de conexión. Otros de los factores que ha hecho que *Amazon API Gateway* esté actualmente teniendo una gran acogida, es que *Amazon API Gateway* puede aumentar su funcionalidad complementándose con el uso de otros productos de la suite AWS, como por ejemplo: *AWS Lambda*, *Amazon Cognito* o *Amazon Kinesis*. Entre las principales características [19] [20] de *Amazon API Gateway* destacan:

- Facilidad en la creación y desarrollo de APIs: Mediante la interfaz gráfica de la consola de *Amazon API Gateway* se pueden crear de una manera rápida y simple las distintas APIs REST y los recursos y métodos vinculados con cada una de ellas. La consola de administración de *Amazon API Gateway* provee la funcionalidad necesaria para el gobierno de los distintos aspectos relativos al ciclo de vida de las APIs.

- Compatibilidad con API REST y API WebSocket: *Amazon API Gateway* permite la creación de API REST tradicionales pero además incluye en su oferta la creación de APIs WebSocket que posibilitan la comunicación bidireccional, ya que con este nuevo enfoque se mantiene una comunicación persistente entre el cliente y el servidor, pudiendo el servidor iniciar la comunicación con el cliente cuando lo estime necesario.

- Resilencia: *Amazon API Gateway* brinda la posibilidad de limitar el número de peticiones por segundo que recibe cada API. En adición, *Amazon API Gateway* posee un mecanismo de caché con tiempo de validez de la información configurable y controles de usuario basado en mecanismos de API Key.

- Gestión del ciclo de vida de las APIs: Con *Amazon API Gateway* podemos desplegar varias versiones de la misma API de manera simultánea, con objeto de asegurar la compatibilidad hacia atrás a los consumidores que lo necesiten. A su vez, existen mecanismos de control de publicación en los que la API evoluciona por distintas fases (alfa, beta y producción).

- Generación de SDK (Software Development Kit): Una de las capacidades más interesantes de *Amazon API Gateway* es la capacidad de generar un SDK cliente para distintos lenguajes para invocar a los distintos recursos de las APIs. Además, desde el cliente SDK generado se pueden administrar las claves de las APIs y las solicitudes de inicio de sesión. De esta manera, *Amazon API Gateway* proporciona a los desarrolladores una integración rápida para invocar y administrar determinados aspectos de sus APIs mediante desarrollos que se hayan realizado en algunos de sus lenguajes de trabajo soportados. *Amazon API Gateway* ofrece la posibilidad de generar SDK clientes para los siguientes lenguajes de programación: Java, JavaScript, Java para Android, Objective-C o Swift para iOS y Ruby.

- Monitorización de las APIs: *Amazon API Gateway* proporciona una serie de cuadros de mandos para la consulta de métricas, volumen de llamadas o tasa de error. Esta característica del producto se adquiere mediante el uso del componente de la suite AWS *Amazon CloudWatch* por lo que no es intrínseca al producto *Amazon API Gateway*.

- Autorización: En lo que respecta al servicio de seguridad de autorización *Amazon API Gateway* posee un conjunto de políticas de acceso que permiten controlar el nivel de granularidad en el acceso a cada API y a

los recursos asociados a cada una de ellas. Al igual que ocurría con la funcionalidad de monitorización los mecanismos de autorización de este sistema API Management también puede ser aumentados mediante la integración con otros componentes de la suite AWS, como *Amazon Identity*, *Amazon Access Management* y *AWS Lambda*.

- Claves de APIs o API Keys: El producto permite crear múltiples claves para cada API y establecer una serie de permisos para cada una de las claves. Mediante esta estrategia se logra que desarrolladores puedan consumir el API en la que estén interesados sin que suponga un problema de seguridad para el propietario del sistema API Management. Otra funcionalidad asociada con la generación de claves de API es la monetización o tarificación, que permite asignar políticas de facturación y planes de cuota de uso del servicio en función de cada cliente.

- Mecanismos de modelado y transformación del mensaje: *Amazon API Gateway* cuenta con una serie de componentes que permiten modificar el mensaje en su paso por el sistema de API Management, posibilitando realizar transformaciones en el formato del mensaje o en su propio contenido.

En la figura puede contemplarse como mediante el uso de distintos componentes de la suite de *Amazon AWS* se aumenta y completa las distintas funcionalidades base del producto *Amazon API Gateway*:

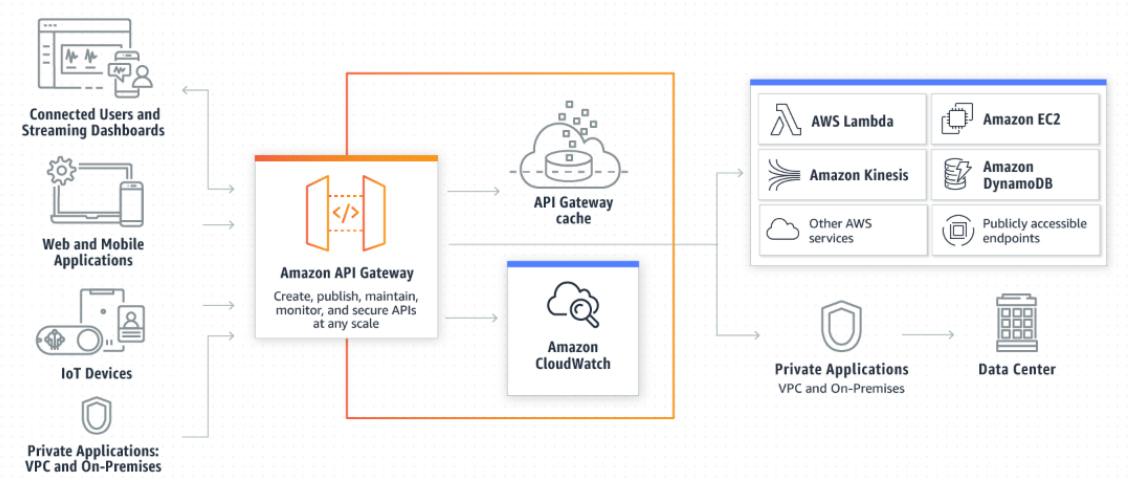



**Figura 7. Componentes suite AWS**

**Fuente: https://aws.amazon.com/es/api-gateway/**

### 4.2.4. Microsoft Azure API Management

*Microsoft's Azure API Management* fue creado originalmente por *Apiphany*, empresa que *Microsoft* adquirió en 2013 para integrarla posteriormente en su plataforma de servicios *Azure* en 2014. *Microsoft's Azure API Management* es una solución cloud de API Management en la que se ofrecen distintos planes de contratación, en función, del nivel de uso que se quiera realizar de esta solución. Al igual que ocurre con el sistema API Management de *Amazon*, la funcionalidad de *Microsoft's Azure API Management* puede incrementarse mediante el uso de componentes de la suite de Microsoft Azure. *Microsoft's Azure API Management* oferta diversas funcionalidades [21] [22] mediante los siguientes tres componentes [23] [24] [25] :

- **API Gateway**: Este componente actúa como un servidor proxy aceptando las distintas llamadas procedentes de los consumidores y realizando distintas tareas de control de acceso y enrutado. Sus principales características son:
    - Aceptación de la petición y enrutamiento de la misma hacia los distintos back-ends.
    - Comprobación de credenciales de seguridad como claves de API, tokens JWT, certificados…
    - Asignación de cuotas de servicios y establecimiento de límites de uso.
    - Edición en caliente de características de las APIs sin necesidad de reiniciar el sistema.
    - Sistema de cacheado de las respuestas de los distintos back-ends. Mediante el uso de esta funcionalidad puede liberarse de carga a los distintos back-ends y al propio sistema API Management.
    - Almacenamiento de los metadatos de las llamadas realizadas por los distintos consumidores para su posterior tratamiento y análisis.
- **Azure Portal**: Es el portal de administración que posibilita, mediante la interacción con una interfaz gráfica, la gestión de las funciones relacionadas con el ciclo de vida de la API. Destacan las siguientes funcionalidades del componente:
    - Mecanismos definición e importación de esquemas de API.
    - Empaquetado de APIs para crear distintos productos.
    - Directivas: Mediante el establecimiento de directivas se pueden configurar funcionalidades como cuotas de servicio,

transformaciones de comunicación, como cambios en el protocolo y en el formato del mensaje…

- Monitorización: Visualización de información de análisis de servicios y métricas de las APIs.

- Administración de usuarios: Selección de mecanismos de autorización y control de acceso para los distintos usuarios de las APIs y del componente Azure Portal.

- **Portal de desarrolladores**: Este componente provee de una interfaz gráfica para ofertar distintas secciones de interés para los desarrolladores. Las principales características del portal de desarrolladores son:

  - Acceso a documentación de cada API generada de manera automática.

  - Acceso a códigos de generación de clientes en distintos lenguajes como JAVA o C# para la implementación de clientes que consuman las APIs de interés.

  - Exposición de logs, bus y problemas con las APIs para que los desarrolladores puedan consultar dicha información.

En la *Figura 8. Componentes Azure API Management* puede contemplarse los distintos componentes explicados con anterioridad que conforman la solución *Microsoft Azure API Management*:

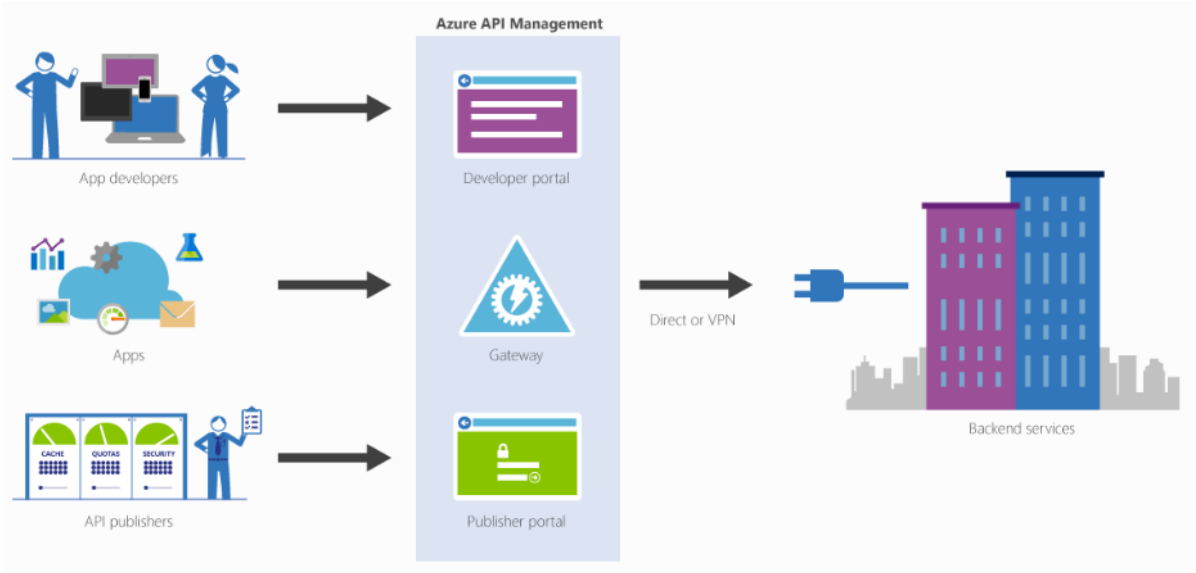



**Figura 8. Componentes Azure API Management**

**Fuente: https://www.kabel.es/azure-api-management/**

### 4.2.5. WSO2 API Manager

La suite de productos *WSO2* oferta multitud de servicios de integración mediante sus distintos componentes como *WSO2 API Manager*, *WSO2 Enterprise Integrator*, *WSO2 Identity Service*…Un aspecto clave a destacar es que de todas las soluciones de sistemas API Management estudiadas en el presente trabajo de fin de master, *WSO2 API Manager* es la única que es open-source. En lo que respecta al componente *WSO2 API Manager*, la empresa WSO2 oferta dos posibles instalaciones: on-premises mediante el producto *WSO2 API Manager* y la posibilidad de instalar el producto en el cloud mediante el producto *WSO2 API Cloud* existiendo la posibilidad de alojar el producto en servidores de *Microsoft's Azure* o en servidores de *Amazon Web Services*. En la *Figura 9. Componentes WSO2 API Manager* se presentan los principales componentes [26] de los que consta la arquitectura del sistema *WSO2 API Manager*:

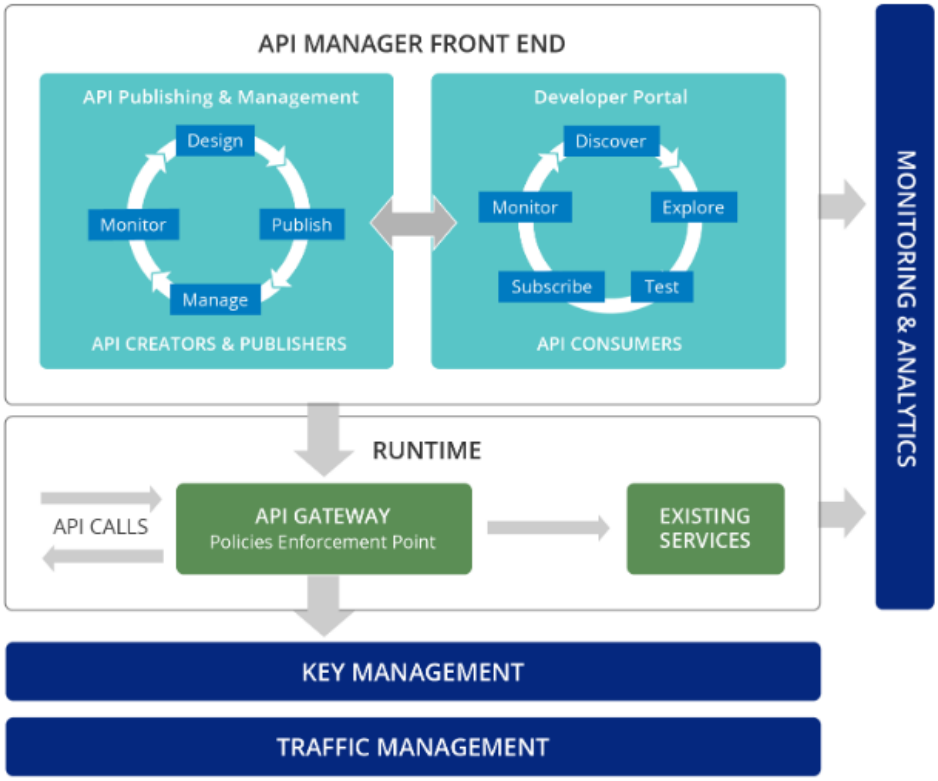



**Figura 9. Componentes WSO2 API Manager**

**Fuente: https://docs.wso2.com/display/AM210/Key+Concepts**

- **API Gateway**: Este componente es un API proxy encargado de securizar, proteger, administrar y escalar las llamadas de las APIs. El componente API Gateway recibe las peticiones entrantes y le aplica las políticas de consumo de uso y de seguridad. En adición, este componente posibilita la administración de las estadísticas de las APIs.

- **Key Manager**: El componente Key Manager actúa como servicio de tokens de seguridad (STS) y es el encargado de emitir los tokens de seguridad a los servidores de recursos para que estos servidores puedan autenticarse y autorizarse de manera correcta frente al sistema API Management.

- **Traffic Manager**: Este componente tiene como misión principal la regulación y administración del tráfico de las APIs. El componente *Traffic Manager* permite que las APIs estén disponibles a los distintos consumidores en diferentes niveles de servicio y con diferentes grados de seguridad. El componente es capaz de procesar las políticas de servicio y seguridad definidas en tiempo real.

- **Analíticas**: El componente *WSO2 API Manager*, mediante su integración con el componente *WSO2 Analytics*, es capaz de proporcionar información de estadísticas en distintos dashboards gráficos que ya se encuentran embebidos en el componente *WSO2 API Manager.* El componente de analíticas permite la monitorización de las APIs pudiendo detectar actividades inusuales o fallos de los servicios, en adición, el componente posibilita el filtrado de estadísticas, como por ejemplo clasificar las estadísticas por información geográfica, para poder realizar un análisis exhaustivo de los registros estudiados.

- **MarketPlace**: Este componente permite a los desarrolladores descubrir APIs y testearlas de manera online antes de consumirlas en sus aplicaciones, además permite realizar operaciones de monetización, consultar el feedback de los usuarios y solicitar mejoras a los administradores de las APIs.

- **Publisher**: Este componente se materializa en una interfaz gráfica que permite a los creadores de las APIs desarrollar, publicar, monetizar, analizar, documentar, escalar y versionar APIs de una manera fácil e intuitiva.

La multitud de componentes anteriores hacen que el componente *WSO2 API Manager* pueda ofrecer la siguiente carta de servicios y funciones [27]:

- Diseño y prototipado de APIs:

    - Diseño de APIs antes de su implementación, siguiendo el enfoque API First Design.

    - Mockeado de APIs.

    - Soporte para protocolos SOAP y REST y para los formatos JSON y XML.

- Publicación y gobierno de APIs:
    - Publicación de APIs de manera interna y externa.
    - Control de visibilidad de las APIs, con posibilidad de restringir el acceso a posibles consumidores.
    - Asignación de API keys en función del entorno de ejecución.
    - Control del despliegue de las APIs por versión.
- Control de acceso y cumplimiento de seguridad:
    - Restricción de acceso a las APIs por dominio o IP.
    - Validación del mensaje para verificar que cumple el esquema definido en el API.
    - Protección de amenazas, detección de bots y de tokens fraudulentos.
    - Acceso a las APIs mediante el uso del protocolo OAuth.
- Funciones para desarrolladores:
    - Búsqueda de APIs por categorías, proveedor y nombre.
    - Adquisición de API keys.
    - Consola interactiva de testing online.
    - Sistema de notificaciones para desarrolladores, informando de nuevos cambios o de liberaciones de nuevas versiones.
- Administración y escalado del tráfico de las APIs:
    - Separación del tráfico de los entornos de ejecución (producción y desarrollo) en diferentes API gateways.
    - Mapeo entre el protocolo HTTP y otros protocolos como JMS.
    - Protección de los back-ends mediante limitación del tráfico que puede dirigirse hacia los mismos.
- Monitorización y monetización:
    - Establecimiento y consulta de distintas analíticas de las APIs relativas a peticiones, respuestas, fallos, subcripciones…

- Visor de logs en tiempo real.
- Esquemas de pago configurables para monetizar el uso de las APIs.
- Monitorización de cumplimiento de SLAs.
- Publicación de eventos propios y creación de dashboards personalizados.

# 5. Análisis de servicios de seguridad

## 5.1. Apigee

### 5.1.1. Autenticación y autorización

En lo que respecta a los servicios de seguridad de autenticación y autorización *Apigee* los clasifica en distintos ámbitos:

- En la administración del sistema de API Management.
- En las fronteras del sistema de API Management:
    - Comunicaciones desde el cliente hacia el API Management (Inbound).
    - Comunicaciones desde el API Management hacia el back-end (Outbound).

En lo que respecta a la autenticación de los usuarios de administración [28] del sistema API Management, *Apigee* ofrece tres posibles mecanismos de autenticación:

- **OAuth2**: El componente *Apigee Edge* permite realizar distintas funciones de administración mediante llamadas, que son autenticadas a través de tokens OAuth2, a la API REST de administración de Apigee. En la *Figura 10. OAuth2 comunicación Apigee* se ilustra el flujo de comunicación inicial entre los distintos agentes involucrados en el proceso de autenticación:

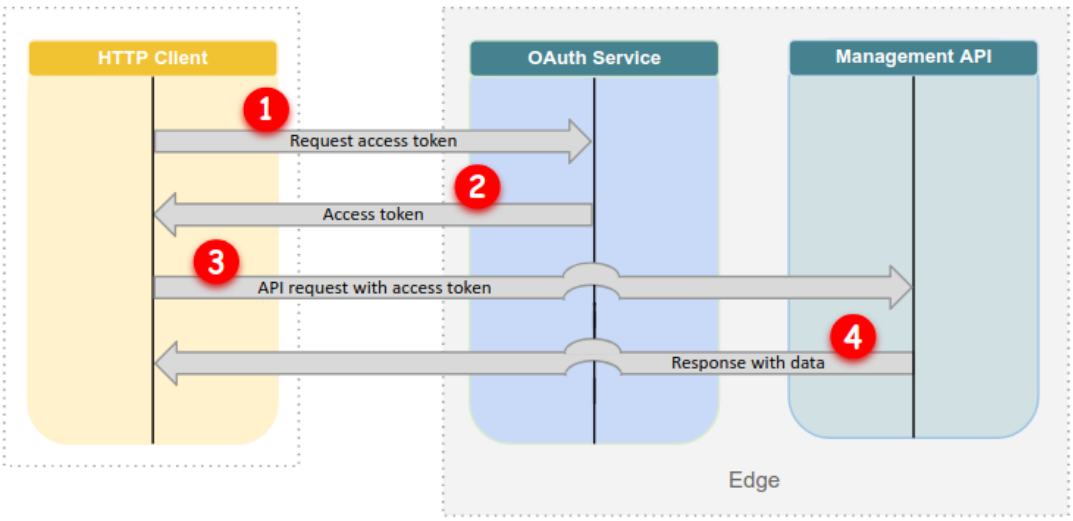



**Figura 10. OAuth2 comunicación Apigee**

**Fuente:https://docs.apigee.com/api-platform/system-administration/using-oauth2**

- **SAML**: Mediante la autenticación con SAML se implementa un entorno de Single Sign On (SSO) con el que poder autenticarse una única vez para acceder a la interfaz gráfica del componente *API Sense*, al API REST de administración de *Apigee* o a cualquier otro servicio que se tenga configurado y que soporte SAML.

- **Basic Authentication**: Este método de autenticación consiste en enviar las credenciales de autenticación del usuario (login/password) en cada petición que se realice al API REST de administración. Es el método menos seguro de los mecanismos de autenticación soportados, ya que no se encriptan ni se hashean los credenciales sino que únicamente se codifican en base64. *Apigee* nos alerta de que no recomienda el uso de este método de autenticación y de que tiene intención de deprecarlo en un futuro.

En adición, *Apigee* ofrece la posibilidad de incrementar la seguridad de los mecanismos de autenticación de administración del sistema anteriores explicados, mediante la inclusión de mecanismos de **autenticación de doble factor**. Este mecanismo puede ser configurado a nivel de usuarios individuales o a nivel de organización.

Una vez que los distintos usuarios se autentiquen de manera correcta tendrá lugar el proceso de autorización, que será el encargado de comprobar si dicho usuarios tiene privilegios para acceder al recurso o realizar la operación solicitada. *Apigee* usa el modelo RBAC (Role Based Access Control) para realizar este proceso. En lo que respecta a la autorización relativa a la administración del sistema de API Management, *Apigee* cuenta con una serie de roles [29] predefinidos:

- Administrador de organización: Super usuario.

- Administrador de la organización de solo lectura: Administrador que únicamente puede consultar y leer los recursos de la organización.

- Administrador de operaciones: Administrador con permisos para realizar despliegues y test sobre las APIs, para el resto de recursos únicamente tiene acceso de lectura.

- Usuario de negocio: Crear y administrar productos que contiene APIs, desarrolladores, creación de información de uso de APIs. Para el resto de recursos tiene acceso de solo lectura.

- Usuario: Permisos para crear API y probarlas en el entorno de desarrollo, para el resto de recursos tiene acceso de solo lectura.

Aunque el producto *Apigee* posee los roles anteriores por defecto, se pueden crear nuevos roles con sus correspondientes permisos asociados, para así aumentar la granularidad en el control de privilegios.

Para la autorización Inbound el componente *Apigee* sigue el mismo modelo RBAC expuesto anteriormente, con la salvedad de que no se tienen en cuenta los roles predefinidos anteriormente, puesto que eran de administración y no de consumo. En lo que respecta a los mecanismos de autenticación y autorización Inbound [30] [31] *Apigee* presenta la siguiente oferta:

- OAuth2.
- LDAP.
- Api Key.
- SAML.
- JWT

En lo que respecta al uso del protocolo OAuth2, *Apigee* realiza una clasificación en distintos niveles de autenticación [32] en función del consumidor:

- Autenticación de usuario de aplicación: Concede privilegios al usuario basado en los roles y permisos que dicho usuario tenga asignado. Para solicitar el token OAuth2 el usuario tendrá que especificar su usuario y contraseña que le fueron asignados con anterioridad.
- Autenticación de aplicación cliente: Concede pleno acceso para realizar peticiones a las APIs una vez el usuario se ha autenticado. Para solicitar el token OAuth2 se tiene que especificar el id de cliente y el literal *client secret* que se obtiene desde la interfaz de administración.
- Autenticación de cliente de organización: Igual que el anterior pero con acceso a las APIs de toda la organización.
- Autenticación de usuario administración: Concede pleno acceso para realizar peticiones a las APIs de las que dicho usuario es administrador, una vez el usuario se ha autenticado. Para solicita el token OAuth2 se tiene que especificar el usuario y password de administrador.

Finalmente en la zona Outbound (comunicaciones Apigee hacia los back-ends), *Apigee* cuenta con los siguientes mecanismos de autenticación [33] para poder iniciar las comunicaciones con los distintos back-ends:

- OAuth2.
- Basic Authorization.
- Api Key.
- SAML.

### 5.1.2. Cifrado del tráfico y encriptación

*Apigee* oferta la posibilidad de usar TLS/SSL [34] para encriptar todas las comunicaciones en las que el sistema API Management participa:

- Comunicaciones desde los consumidores de APIs hacia el sistema API Management, con independencia de que sean APIs de negocio o la propia API de administración del producto.

- Comunicaciones con los distintos back-ends de servicios.

*Apigee* también ofrece la posibilidad de incorporar mutual-ssl, escenario en el que la comunicación es cifrada en ambos extremos de la comunicación.

En el ámbito de la protección de la información crítica o sensible [35] *Apigee* cuenta con un mecanismo de enmascaramiento y con otro mecanismo de cifrado:

- Enmascaramiento y ocultación: Mediante esta funcionalidad se puede ocultar o enmascarar información sensible que viaje en los mensajes como por ejemplo información bancaria, de esta manera, esta información crítica no aparecerá en los ficheros de logs o aparecerá enmascarada mediante asteriscos.

- Key Value Maps: El mecanismo *Key Value Map* ofertado por *Apigee* es un almacén que guarda información cifrada en forma de mapas clave valor. De esta manera, *Apigee* garantiza el almacenado de información sensible de manera segura, sin que los datos queden expuestos.
  Existen distintos scopes o ámbitos de almacenamiento (*organization*, *enviroment* y *apiproxy*), en función, de la tipología de la información sensible a almacenar tendrá como destino un Key Value Map u otro. *Apigee* también oferta la posibilidad de usar los Key Value Maps sin encriptación, ya que podría ser útil para disponer de un almacén compartido entre diversos recursos del sistema de API Management. En la *Figura 11. Key Value Maps Apigee* se muestra un ejemplo de la estructura de dos Key Value Map con y sin encriptación:

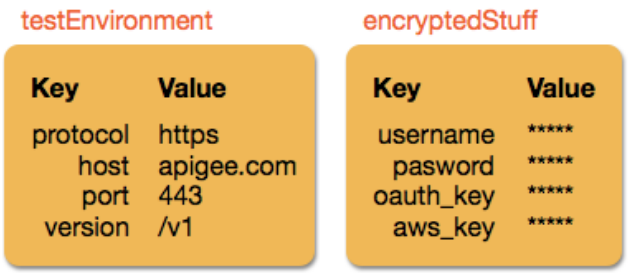



**Figura 11. Key Value Maps Apigee**

**Fuente: https://docs.apigee.com/api-platform/cache/key-value-maps.html**

### 5.1.3. Alertas y monitorización

*Apigee* dispone de una API de monitorización que permite la detección de problemas de rendimiento y el rápido diagnóstico de incidencias. La funcionalidad de esta API de monitorización se puede consumir haciendo uso del protocolo REST mediante comunicaciones petición-respuesta o mediante un dashboard para la administración de monitorización con el que cuenta el producto *Apigee*. Se ofertan las siguientes funcionalidades de monitorización [36]:

- Monitor de actividad reciente.
- Identificación de picos de sobrecarga.
- Investigación de incidencias mediante estudio de la latencia y del código de estado HTTP de la comunicación.
- Configuración de alertas y notificaciones.
- Elaboración de diferentes tipos de informes en función de la información objeto de estudio.

En lo que respecta a la notificación de alertas, *Apigee* oferta diversos canales de notificación como: Email, Webhook, PagerDuty y Slack.

Recientemente *Apigee* oferta en sus releases Alpha, un nuevo servicio de seguridad. Además de proporcionar informes de monitorización, *Apigee* también es capaz de proporcionar informes de seguridad [37]. Mediante estos informes de seguridad se puede obtener en tiempo real una perspectiva de seguridad de la configuración del sistema y del tráfico de comunicaciones. En los informes de seguridad puede consultarse la siguiente información:

- Tráfico Inbound y Outbound.
- Tráfico que usa el protocolo HTTPS y HTTP.
- Consulta del grado de protección de cada API.
- Porcentaje de tráfico que cada API consume.
- Porcentaje de tráfico que consume cada desarrollador.
- Porcentaje de tráfico que se envía a cada back-end.

### 5.1.4. Protección frente amenazas

*Apigee* expone el grueso de los servicios de seguridad de su oferta mediante lo que se denomina en el producto como políticas. Las políticas pueden aplicarse en cualquier dirección del flujo del mensaje (Inbound y Outbound) y son funciones que enriquecen la comunicación añadiendo una nueva funcionalidad, ya sea de seguridad o no. En la *Figura 12. Políticas y flujos Apigee* se representa el concepto y el uso de las políticas en los diferentes flujos de comunicaciones que se producen en el sistema de API Management:

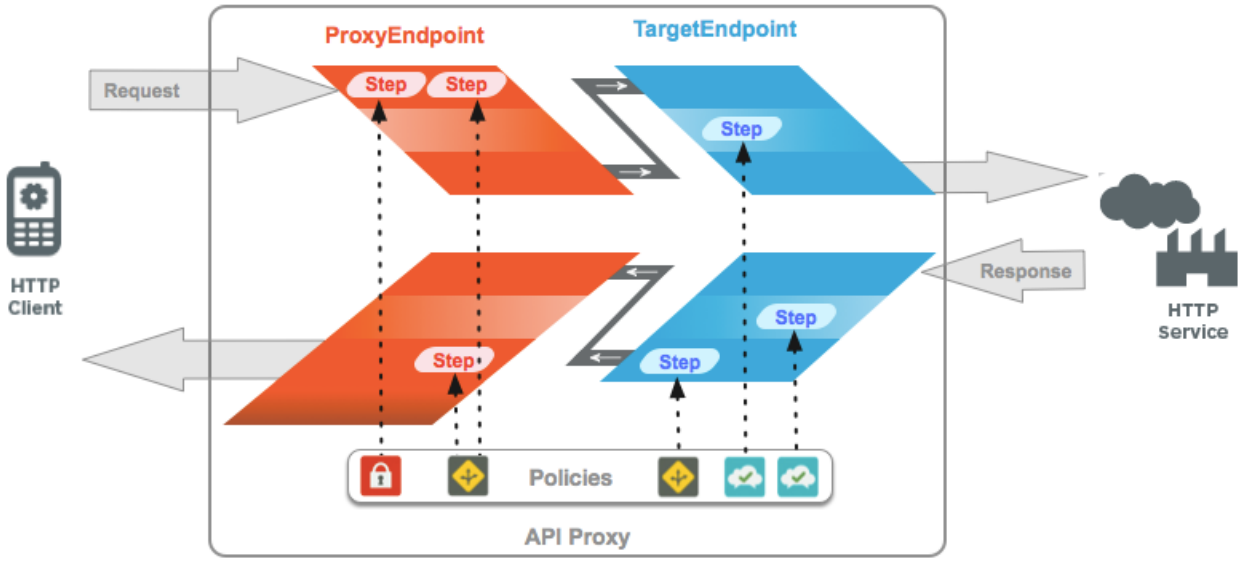



**Figura 12. Políticas y flujos Apigee**

**Fuente: https://docs.apigee.com/api-platform**

*Apigee* ofrece los siguientes mecanismos de seguridad [38] en lo que respecta al ámbito de la prevención de amenazas:

- Protección contra ataques JSON y XML: Los ataques JSON y XML atacan los parseadores de estructuras que se usan en el proceso de serialización y deserialización para afectar a la calidad del servicio e inducir ataques de denegación de servicio. *Apigee* cuenta con las políticas *XML Threat Protection* y *JSON Threat Protection* para evitar esta tipología de ataques.

- Protección del contenido del mensaje en general: La mayoría de los ataques basados en el contenido del mensaje utilizan cabeceras HTTP, parámetros usados en las peticiones o contenido en el mensaje para lograr la ejecución de código. Un ejemplo de estos ataques son los ataques de inyección SQL. *Apigee* cuenta con la política *Regular Expression Protection* para evitar este tipo de ataques.

### 5.1.5. Control y regulación del tráfico

Para asegurar el rendimiento y la disponibilidad del sistema es crucial definir unos umbrales de consumo de tráfico para cada consumidor y asegurar que ninguno de estos consumidores rebasa los límites de tráfico definidos. Estos mecanismos [39] son una forma de protección frente a ataques de denegación de servicio. *Apigee* oferta estos mecanismos mediante el uso de las políticas *Spike Arrest*, *Quota* y *ConcurrentRateLimiting*:

- *Spike Arrest*: Esta política permite definir un límite de peticiones para la API. El límite definido debe de ser un valor que coincida con la capacidad máxima que el back-end asociado a esa API está dispuesto a soportar. Es posible definir límites como por ejemplo 30 peticiones por minuto, con el fin de evitar ráfagas de tráfico repentinas causadas por usuarios malintencionados o atacantes que buscan la denegación del servicio.

- *Quota*: Mecanismo que establece una limitación del número peticiones que un cliente puede realizar a una API mediante el uso de un contador distribuido a nivel de cliente.

- *ConcurrentRateLimiting*: Esta política impone un límite en el número de solicitudes, para un tiempo determinado, que los servicios de una API pueden enviar a sus correspondientes servicios asociados al backend.

## 5.2. CA API Management

### 5.2.1 Autenticación y autorización

La gran mayoría de los servicios de seguridad relacionados con la autenticación y la autorización en el sistema *CA API Management* se concentran en su componente *CA API Gateway* que permite la inserción y composición de distintas políticas de seguridad mediante su componente *Policy Manager*. Existen multitud de mecanismos de autenticación y autorización ofertados [40] por el componente *CA API Management*:

- Oracle Access Manager: Se delega la autenticación y la autorización en la solución *Oracle Access Manager Server*.

- Autenticación mediante un proveedor de identidad: Autentica los credenciales que provienen de un mecanismo de autenticación determinado contra un proveedor de identidades. Este mecanismo no incluye el servicio de autorización y es útil cuando se quiere lograr un fuerte desacoplamiento entre los mecanismos de autenticación y autorización.

- Autenticación y autorización con proveedor de identidad: Se delega la autenticación y la autorización en el proveedor de identidad

seleccionado. *CA API Management* acepta los siguientes tipos de proveedores de identidad: LDAP Identity Providers, Simple LDAP Identity Providers, Federated Identity Providers (FIP) e Internal Identity Providers (IIP). A su vez este mecanismo admite la siguiente tipología de credenciales: HTTP Basic Credentials, SAML Token Profile y transporte SSL y TLS.

- Autenticación mediante servidor *Radius*: En este mecanismo *CA API Management* solo ofrece el servicio de seguridad de autenticación y no el de autorización.

- Autenticación mediante CA Single Sign-On: Autenticación contra el servicio de SSO de la propia compañía. Este método admite dos tipologías de credenciales: usuario/password y certificado X.509. CA permite delegar la autenticación y la autorización en el componente *CA Single Sign-On*.

- Autenticación con *Tivoli Access Manager*: La autenticación y la autorización se delegan en la solución propietaria de IBM *Tivoli Access Manager*.

- Autenticación basada en el estándar OASIS WS-Security: CA soporta todos los tipos de credenciales que soporta el estándar WS-Security para realizar la autenticación.

- Autenticación y autorización mediante SAML.

- Autenticación mediante extracción de atributos de certificado X.509. Esta política debe ser combinada con otras como transporte SSL/TLS, WS-Security, SAML o autenticación con proveedor de identidad.

- Autenticación mediante consulta a *Cassandra*: El proceso de autenticación y autorización se delega en el resultado de la ejecución de consultas en la base de datos de *Cassandra*.

- Autenticación mediante consulta a través del protocolo JDBC: El proceso de autenticación y autorización se delega en el resultado de la ejecución de consultas en una base de datos mediante el uso del protocolo JDBC.

- Autenticación contra FTP: *CA API Management* oferta este mecanismo de autenticación y autorización para acceder a servidores FTP (File Transfer Protocol).

- Basic Authentication: Autenticación mediante la cabecera HTTP *Authorization* que contiene el usuario y password codificados en Base64.

- Comprobación de Cookies: *CA API Management* oferta este mecanismo para reforzar la seguridad en el proceso de autenticación. El mecanismo exige que exista cierto valor para una Cookie HTTP determinada o que simplemente exista dicha Cookie.

- Autenticación mediante hashes *NTLM*: Se usan credenciales de tipo *NTLM* en el proceso de autenticación y autorización. Las credenciales *NTLM* contienen la siguiente información: nombre completo del usuario, directorio del usuario, permisos de la cuenta del usuario y grupos a los que pertenece el usuario.

- Autenticación contra identidad en dominio remoto: Permite a un usuario autenticarse contra un dominio remoto de *Windows Active Directory*.

- Autenticación mediante el uso del protocolo SSH.

- Autenticación mediante TLS/SSL con posibilidad de requerir el certificado al cliente (mutual SSL).

- Autenticación en Windows mediante el uso del protocolo Kerberos.

- Autenticación mediante usuario y contraseña con expresiones XPATH.

- Autenticación mediante JSAM: La autenticación y la autorización se delegan en la solución propietaria de *SUN Java System Access Manager*.

- Autenticación de doble factor.

En adición *CA API Management* puede actuar como proveedor de identidad SAML, como servidor de autorización OAuth y como servidor OpenID Connect.

El componente *CA API Management* además de proveer los mecanismos de autenticación y autorización anteriormente expuestos es capaz de brindar mecanismos de autenticación y autorización ad-hoc para aplicaciones móviles [41] mediante el componente *CA Mobile API Gateway*:

- Políticas de autenticación basadas en geolocalización.

- Single Sign On para aplicaciones móviles.

- Autenticación One Time Password (OTP) para soportar autenticación multi-factor. La contraseña OTP es una cadena numérica o alfanumérica de caracteres que identifica al usuario una vez se haya autenticado mediante el uso de sus credenciales, proporcionando un factor extra de seguridad. La contraseña OTP puede ser enviada por correo o vía SMS.

- Mecanismos de autenticación basados en proximidad:

    - Código QR.

    - Near Field Communication (NFC).

- Bluetooh Low Energy (BLE).

- Certificados SSL de confianza y SSL Pinning:

  - Autenticación del servidor mediante certificados de confianza: Este mecanismo provee un nivel de seguridad mínimo, durante el proceso de negociación SSL, el cliente móvil evalúa el certificado que el servidor le ha mandado contra la lista de certificados raíz de confianza almacenada en el dispositivo móvil. SI el certificado no se encuentra firmado por una CA de confianza, la conexión es rechazada.

  - Autenticación mediante SSL Pinning: Es el método basado en certificados más seguro de los que ofrece *CA API Management*. El cliente móvil únicamente permitirá conexiones SSL de servidores que presenten certificados con valores de campos que coincidan exactamente con los almacenados por el dispositivo móvil, como por ejemplo el campo fecha de expiración. En definitiva, este método de autenticación del servidor se basa en realizar un matching de los valores de los campos del certificado presentado por el servidor, con valores de campos relativos a ese certificado, que el cliente móvil ya tiene almacenados con anterioridad.

  - Autenticación mediante clave pública SSL Pinning: Mecanismo de autenticación homólogo al anterior pero el campo que se comprueba es la clave pública del certificado presentado por el servidor. Para que se pueda realizar la conexión SSL esta clave pública tiene que coincidir con la clave pública que el cliente móvil tiene almacenada para ese certificado.

La *Figura 13. CA API Gateway autenticación móvil* nos ilustra los mecanismos de seguridad que CA oferta para las aplicaciones basadas en movilidad:

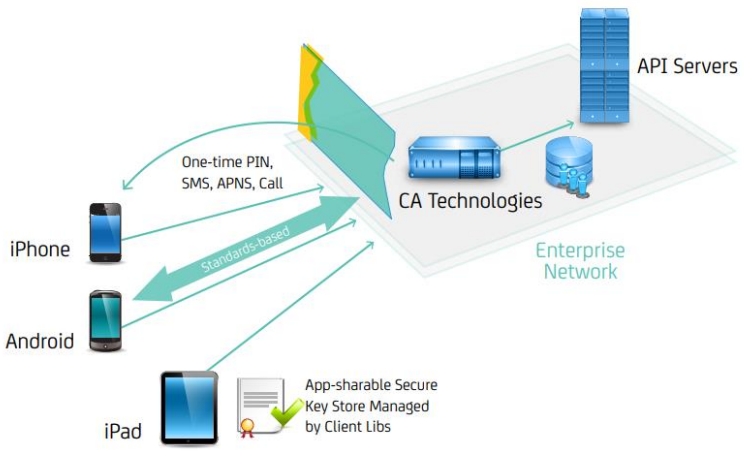



**Figura 13. CA API Gateway autenticación móvil**

**Fuente: https://www.ca.com/content/dam/ca/us/files/white-paper/identity-in-mobile-security.pdf**

En lo que respecta a la administración de la plataforma *CA API Management* al igual que *Apigee* usa un mecanismo de autorización basado en RBAC para la gestión interna del componente *CA API Gateway.* CA oferta múltiples vías para la administración del producto *CA API Gateway*: API REST, API SOAP e interfaz gráfica web.

### 5.2.2. Cifrado del tráfico y encriptación

*CA API Management* posibilita el cifrado [42] de todas las comunicaciones desde/hacia el sistema API Management mediante el uso de SSL/TLS, pudiendo incluir la solicitud del certificado del cliente para realizar comunicaciones mutual SSL, en caso de ser requeridas. En adición *CA API Management* cuenta con otro mecanismo de seguridad para la capa de transporte consistente en la obtención del certificado de una sesión XMPP (protocolo usado en mensajería instantánea), para usar este último mecanismo la sesión XMPP tiene que usar TLS en sus comunicaciones.

En cuanto al tratamiento de la información sensible o crítica al igual que Apigee ofrece dos funcionalidades:

- Ocultamiento de datos sensibles [43] en auditoría: Se enmascara la información sensible de los distintos tratamientos del mensaje realizados en las distintas fases y procesos del mensaje que realiza el componente *CA API Gateway*, así como en los ficheros logs. Para lograr esta funcionalidad *CA API Gateway* usa la política de auditoría de filtrado de mensajes (*AMF policy*).

- Uso de contraseñas almacenadas: *CA API Gateway* proporciona el mecanismo *Stored Passwords* [44] que permite almacenar información crítica de manera cifrada. Las contraseñas almacenadas se cifran usando AES-256 y si además se ha elegido el plan del producto que incluye el módulo de seguridad hardware (*HSM*), a la contraseña se le aplicará un nivel de protección adicional post-cifrado.

### 5.2.3. Alertas y monitorización

En el ámbito de la monitorización *CA API Management* realiza la siguiente clasificación [45] entre los mensajes de auditoría:

- Mensajes de monitorización del sistema: Mensajes que el administrador no puede controlar. Estos mensajes so generados constantemente en segundo plano por el API Gateway y suelen estar relacionados con tareas como inicio del servidor, actualización de licencias, conexión con endpoints JMS…

- Mensajes de monitorización administrativos: Estos mensajes se producen cuando una acción administrativa es llevada a cabo mediante el componente *Policy Manager* o mediante el uso de alguna de las APIs de administración que el producto oferta.

- Mensajes de monitorización de política: Mensajes de monitorización que se generan cuando una política es procesada.

*CA API Management* permite monitorización de diversas fuentes y métricas en el proceso de auditoría que realiza en tiempo real como: memoria acumulada, campos personalizados en el mensaje usado en las comunicaciones, captura de identidad (IP, identificador de usuario…), captura de respuestas erróneas.

En la siguiente figura se muestra a modo de ejemplo un dashboard de monitorización usando el componente *Policy Manager*:

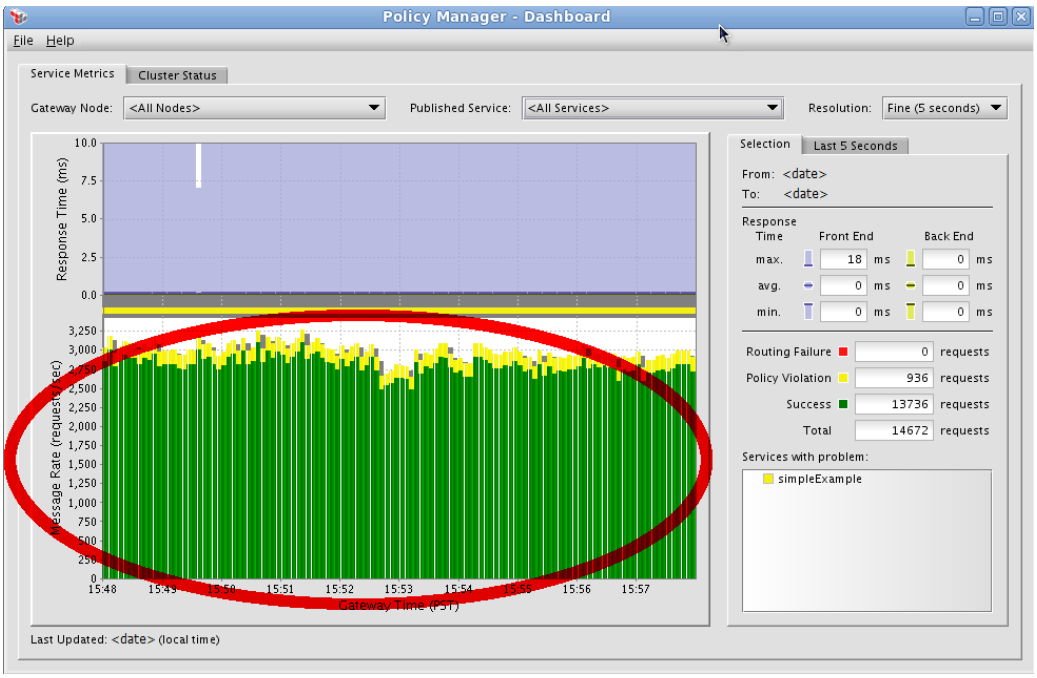



**Figura 14. CA API Gateway Monitorización**

**Fuente: https://docops.ca.com/ca-api-gateway/9-3/en/reference/monitor-the-ca-api-gateway/**

Finalmente en lo que respecta a los canales de envío de las notificaciones soporta dos canales: correo electrónico y mediante *traps* del protocolo SNMP (Simple Network Management Protocol).

## 5.2.4. Protección frente amenazas

*CA API Management* cuenta con distintos mecanismos que realizan determinadas comprobaciones con objeto de incrementar la protección contra diversos ataques y amenazas. A continuación se presentan los distintos mecanismos [46] de protección:

- Limitación del tamaño del mensaje: Este mecanismo permite especificar un tamaño límite para el mensaje enviado en las comunicaciones, cuando el tamaño del mensaje de una petición supere el límite establecido, el componente *CA API Gateway* procederá a rechazarlo.

- Protección contra la inyección de código: Este mecanismo ayuda a blindar la ruta de la URL, parámetros indicados en la URL y el cuerpo del mensaje. Protege frente a ataques de tipo:

    - HTML/JavaScript Injection (XSS).

    - Hex/Octal Encoded HTML/JavaScript Injection.

    - PHP Code Injection—Eval injection.

    - Shell Injection.

    - LDAP Injection.

    - Xpath Injection.

- Protección contra ataques CSRF (Cross-Site Request Forgery): Dentro de esta funcionalidad se ofertan dos mecanismos para incrementar la protección frente a este tipo de ataques:

    - Double Submit Cookie Validation: Se comprueba que el contenido de alguna cookie HTTP contenga algún identificador de sesión que concuerde con el identificador de sesión enviado como parámetro en la petición.

    - HTTP Referer validation: La validación de esta cabecera HTTP puede ser usada para garantizar que el valor del referido está dentro de una whitelist. Aunque de esta cabecera HTTP es fácilmente suplantable, la validación de la misma reduce los vectores de ataque CSRF.

- Protección contra ataques que actúan sobre la estructura del mensaje XML: Posibilidad de especificar el tamaño máximo del mensaje XML contenido en las peticiones entrantes para protegerse contra ataques de denegación de servicio XML que aprovechan debilidades vinculadas con el desbordamiento del tamaño de ficheros. Cuando el mensaje supere el límite de tamaño *CA API Gateway* bloqueará y rechazará el mensaje.

- Protección contra ataques que actúan sobre la estructura del mensaje JSON: Posibilidad de especificar el formato en cuanto a estructura que deberá tener el mensaje JSON procedente de las peticiones entrantes para protegerse contra ataques de denegación de servicio JSON.

- Protección contra ataques de repetición del mensaje: Para combatir estos ataques el componente *CA API Gateway* cuenta con un

mecanismo de comprobación basado en el identificador *replay ID*, mediante el cual es capaz de detectar mensajes repetidos.

- Protección frente ataques de inyección SQL: *CA API Gateway* puede realizar comprobaciones en los distintos componentes de la petición entrante, basadas en patrones de caracteres específicos o palabras clave, que están potencialmente relacionadas con los ataques de inyección SQL.

- Escaneado de la petición usando el protocolo ICAP: Posibilidad de delegar el escaneado de la petición entrante en algún antivirus que soporte el protocolo ICAP, como por ejemplo McAfee, Sophos, Symantec…

- Validaciones de peticiones Open Data (OData): El componente *CA API Gateway* contiene mecanismos para validar el mensaje de las peticiones que usen el protocolo *Open Data* mediante el uso del *Service Metadata Document* (SMD). Se realizan comprobaciones sobre la URI, la consulta y el payload con objeto de verificar que dichos elementos siguen la estructura esperada conforme a la especificación OData v2.0.

- Validación del contenido de la cabecera HTTP *Content-Type*: Se comprueba que el tipo de contenido enviado en la petición entrante concuerde con el tipo de contenido que el componente *CA API Gateway* espera, en caso de no cumplirse con el tipo de contenido esperado, el mensaje se descarta.

### 5.2.5. Control y regulación del tráfico

El sistema *CA API Management* dispone de una serie de herramientas para actuar sobre el tráfico entrante/saliente y aplicar diversas acciones de control sobre el mismo para incidir sobre aspectos como el acceso, la disponibilidad, el caudal y restricciones. Al igual que ocurre con el resto de servicios de seguridad expuestos en los sub-apartados anteriores, los servicios de seguridad de control del tráfico también se concentran en el componente *CA API Gateway*, destacando los siguientes mecanismos [47]:

- Limitación de frecuencia del tráfico: Este mecanismos permite limitar el ratio de transacciones que el componente *CA API Gateway* recibe por parte de un cliente determinado. Este control puede realizarse mediante nombre de usuario, dirección IP, u otro identificador definido. Cuando el límite establecido para un cliente es alcanzado el componente *CA API Gateway* puede empezar a disminuir el número de peticiones (mediante descartes o mediante la aplicación de otras estrategias) o puede empezar a retrasar las peticiones hacia el usuario hasta que el límite sea alcanzado de nuevo. En adición, este mecanismo soporta el establecimiento de un límite de peticiones concurrentes por parte de un cliente, para así evitar que un cliente determinado monopolice los

recursos del componente *CA API Gateway*.

- Establecimiento de cuotas de servicio: Mediante las cuotas de servicio es posible definir el número de peticiones permitido en un rango de tiempo específico por parte de un cliente.

- Limitación de disponibilidad en el tiempo: Es posible definir un rango temporal específico para hacer que un servicio no esté disponible.

- Restricción por rango de direcciones IP: Posibilita la denegación o el acceso a un servicio determinado basado en la dirección IP del servicio destino o en la dirección IP del cliente que realiza la petición entrante en el componente *CA API Gateway*.

## 5.3. Amazon API Gateway

### 5.3.1. Autenticación y autorización

*Amazon API Gateway* es el sistema de API Management que forma parte de la suite de productos de *Amazon Web Services* (AWS). Debido al gran número de productos alojados en la suite AWS, Amazon persigue que la mayoría de productos interactúen entre sí, para conseguir un aumento de la funcionalidad de cada producto mediante la agregación de componentes de la suite en función de las necesidades de cada usuario de la plataforma AWS. Este hecho hace que el producto *Amazon API Gateway* por si solo quede limitado en cuanto a funcionalidad respecto a sus productos competidores que ofertan toda la funcionalidad en un solo producto sin tener que recurrir al uso de componentes adicionales. De esta manera, tal y como puede verse en la *Figura 15. Comunicaciones en Amazon API Gateway* *Amazon API Gateway* se apoya en distintos componentes de la suite como *Amazon Lambda*, *Amazon EC2*, *Amazon S3*… para completar sus funcionalidades de sistema de API Management, entre las que se encuentran los servicios de seguridad.

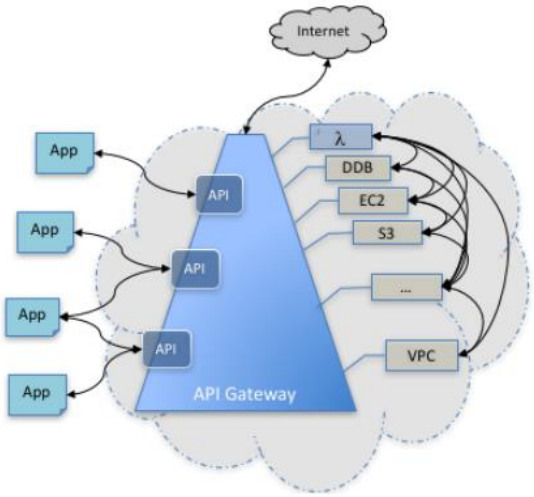



**Figura 15. Comunicaciones en Amazon API Gateway**

**Fuente: https://docs.aws.amazon.com/**

En lo relativo a los servicios se seguridad y autorización *Amazon API Gateway* provee los siguientes mecanismos [48]:

- **Políticas de recursos**: Estas políticas son asociadas a un API para controlar si una entidad (un usuario o un rol creado mediante el componente *Amazon Identity and Access Management*) puede invocar al API en cuestión. Las políticas de recursos permiten realizar el control y autorización mediante:

    - Usuarios que posean una cuenta de Amazon AWS, en este caso se utilizará el protocolo de Amazon *AWS Signature Version 4.*

    - Intervalos de direcciones IP.

    - Nubes privadas virtuales (VPC). Las VPC son redes virtuales alojadas en la nube de AWS y definidas mediante el componente *Amazon Private Cloud*.

- **Funciones y políticas IAM**: *Amazon API Gateway* extiende sus mecanismos de control de acceso mediante el uso de funciones y políticas del componente *Amazon Identity and Access Management (IAM)*. A través del producto *Amazon Identity and Access Management* se administran las distintas identidades dentro de la suite AWS. De esta forma, es posible delegar la autenticación y autorización desde *Amazon API Gateway* hacia el componente *Amazon IAM*, para así controlar las invocaciones realizadas por las distintas entidades hacia las APIs alojadas en el sistema. *Amazon IAM* es el componente que actúa como proveedor de identidades y es compatible con los protocolos *OpenID Connect* (OIDC) y *SAML 2.0.*

- **Autorizadores Lambda**: Amazon *API Gateway* permite delegar la autenticación en una función alojada en el producto *AWS Lambda*. Cuando una entidad realiza una petición a *Amazon API Gateway*, este redirige la petición al componente *Amazon Lambda* que toma la identidad del cliente como entrada y devuelve como salida una política del componente *Amazon IAM*. Mediante este mecanismo se pueden crear métodos de autorización personalizados, existen dos tipos de autorizadores Lambda:

    - Basados en tokens: JSON Web Token (JWT) o OAuth2.

    - Basados en parámetros de solicitud: Este es el autorizador más versátil ya que permite autenticar al usuario mediante la comprobación de cabeceras HTTP u otros parámetros de la petición.

- **Grupo de usuarios de Amazon Cognito**: El resto servicios de seguridad de autenticación y autorización se delegan en el componente de la suite AWS *Amazon Cognito*. Tal y como se muestra en la *Figura 16. Autenticación mediante AWS Cognito* ,el componente *Amazon*

*Cognito* permite los siguientes mecanismos de autenticación:

- Autenticación mediante redes sociales: Siendo compatible con *Facebook*, *Google* y *Amazon*.

- Autenticación mediante proveedor de identidad: Aceptando los protocolos de *OIDC* y *SAML*.

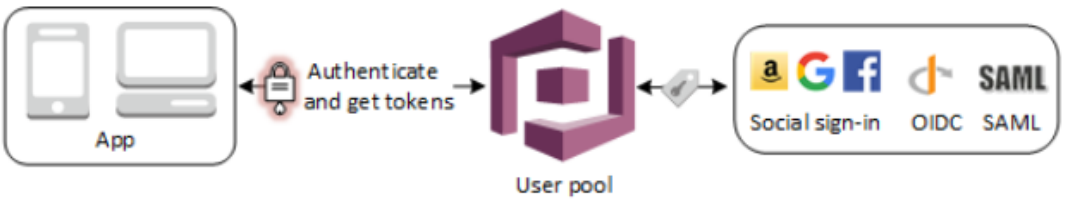



**Figura 16. Autenticación mediante AWS Cognito**

**Fuente: https://docs.aws.amazon.com/**

- **Mutual SSL**: El producto Amazon *API Gateway* permite generar un certificado del cliente para sí mismo, con objeto de que los distintos back-ends puedan verificar que la petición entrante procede del sistema de API Management. Este mecanismo solo se oferta en el lado outbound de las comunicaciones (Amazon *API Gateway* hacia los back-ends).

- **Protección mediante firewall**: Dentro de la suite encontramos el componente Amazon WAF (Web Application Firewall), en el que es posible delegar el control del tráfico entrante en el componente Amazon *API Gateway,* para así aceptar o denegar el tráfico procedente de ciertas entidades en función de las políticas establecidas.

- **API Keys**: Las claves de API son cadenas alfanuméricas que se distribuyen a los clientes de las APIs para concederles acceso a las mismas. Las API Keys no solo representan un mecanismo de control de acceso y autenticación sino que además están estrechamente relacionadas con el proceso de tarificación y la definición de planes de uso por cliente, ya que es posible asignar un plan de uso personalizado por API Key.

### 5.3.2. Cifrado del tráfico y encriptación

En lo que respecta al cifrado del tráfico [49] Amazon *API Gateway* únicamente permite la publicación de APIs mediante el protocolo HTTPS y tampoco permite comunicaciones con back-ends que no usen el protocolo HTTPS, por lo que todas las comunicaciones desde/hacia el sistema API Management estarían cifradas por imposición del fabricante. A su vez, como se comentó en apartado anterior, también es posible generar un certificado para el componente Amazon

*API Gateway* para que el tráfico de las peticiones salientes que el sistema de API Management realiza a los distintos back-ends esté cifrado.

En el ámbito de la encriptación de contraseñas e información sensible, *Amazon* oferta este este mecanismo mediante el componente *Amazon CloudFront* y su característica de cifrado a nivel de campo mediante RSA [50].

En la *Figura 17. CloudFront encriptación* se muestra un ejemplo de encriptación de un número de teléfono ubicado en un campo de la petición entrante que realiza un usuario:

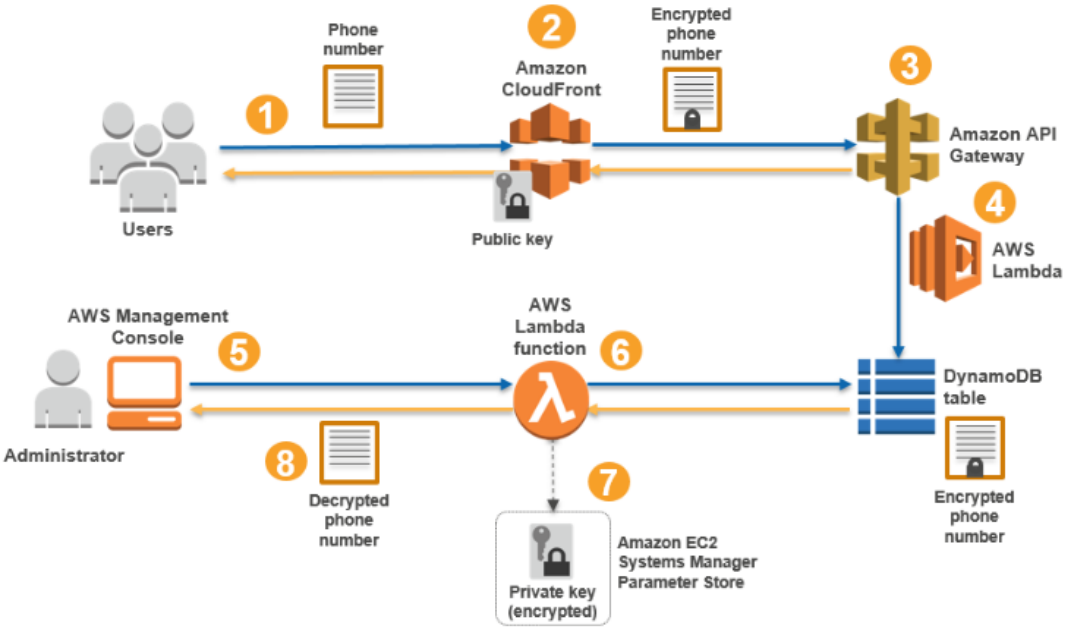



**Figura 17. CloudFront encriptación**

**Fuente: https://aws.amazon.com/**

## 5.3.3. Alertas y monitorización

Nuevamente *Amazon API Gateway* vuelve a delegar servicios de seguridad en otros componentes de la suite AWS: *Amazon X-Ray, Amazon CloudWatch y Amazon CloudTrail.* Los aspectos de monitorización [51] se encuentran distribuidos entre los dos componentes, el producto *Amazon X-Ray* es el encargado de tracear las comunicaciones y *CloudWatch* es el responsable de la generación de métricas.

Mediante el componente *X-Ray* es posible realizar un seguimiento de los problemas de latencia que puedan sufrir las APIs, así como, monitorización de peticiones en función de distintos parámetros de búsqueda como protocolo y diversos campos de la petición.

La generación e interpretación de métricas se aloja en el componente *CloudWatch* que recopila y procesa los datos en crudo enviados por *Amazon API Gateway* cada minuto. Alguna de las métricas de interés en el ámbito de las APIs que oferta *CloudWatch* son los siguientes:

- Tiempos de respuesta del backend.
- Tiempos de respuesta de las APIs.
- Número de errores de cliente por periodo de tiempo.
- Número de errores del servidor por periodo de tiempo.
- Número de solicitudes atendidas por la cache de la API en un tiempo específico.
- Número total de solicitudes entrantes en una API por intervalo de tiempo.
- Latencia: Tiempo transcurrido entre que *Amazon API Gateway* recibe la solicitud del cliente y le devuelve la respuesta al mismo.

El componente *Amazon CloudTrail* se encarga de monitorizar las distintas acciones de administración que son llevadas a cabo por los usuarios del componente *Amazon CloudTrail.*

En lo que respecta a la funcionalidad de alarmas, eventos y logs, reside de manera completa en el componente *CloudWatch*. El componente posee una gran definición de alarmas configurables relativas tanto a métricas de APIs, como a métricas del sistema (CPU, RAM…) ofertando la posibilidad de notificación por correo electrónico cuando la alarma en cuestión se active.

### 5.3.4. Protección frente amenazas

El componente *Amazon API Gateway* no cuenta con ningún sistema propio de protección frente a ataques comunes a las aplicaciones web, sin embargo, usa el componente *AWS Web Application Firewall* (AWS WAF) para lograr dar soporte a estos propósitos.

*AWS WAF* cuenta con mecanismos para hacer frente a ataques que persiguen obtener el control, causar un funcionamiento indeseado o robar información de los sistemas. *AWS WAF* cuenta con herramientas [52] para combatir los siguientes ataques:

- SQL Injection.
- Cross-site scripting (XSS).

- Escáneres de vulnerabilidades: Estos escáneres activos pueden ser usados por un atacante para detectar vulnerabilidades en el sistema de API Management para explotar posteriormente mediante algún ataque.

- HTTP floods: Ataques masivos de peticiones HTTP que buscan causar una denegación del servicio.

- Bots y scrappers.

### 5.3.5. Control y regulación del tráfico

Con objeto de evitar que las APIs se colapsen y para asegurar el correcto funcionamiento del sistema, *Amazon API Gateway* cuenta con mecanismos [53] capaces de limitar las llamadas realizadas por los clientes a las APIs en base a distintos parámetros:

- Control de ráfaga: Limitación del número peticiones que puede recibir un API por segundo.

- Limitación de número peticiones de un API en un periodo de tiempo determinado.

- Limitación basada en plan de uso definida en *Amazon API Gateway*.

Estas limitaciones pueden aplicarse a usuarios, a métodos HTTP del API o incluso a nivel global para todas las APIs alojadas en el sistema de API Management.

## 5.4. Microsoft Azure API Management

### 5.4.1. Autenticación y autorización

El sistema *Microsoft Azure API Management* utiliza elementos denominados directivas, para brindar los servicios de seguridad disponibles en el producto. Al igual que ocurría con el sistema de API Management de *Amazon*, *Microsoft Azure API Management* se apoya en otros productos o servicios de la suite de Microsoft *Azure*, para expandir la funcionalidad de su sistema de API Management.

*Microsoft Azure API Management* cuenta con la siguiente oferta de mecanismos de autenticación y autorización [54] [55] para las entidades que consuman las APIs del sistema:

- **Autenticación básica**.

- **Autenticación mediante certificado de cliente**: Se utiliza el certificado digital del cliente para autenticar al mismo.

- **API Keys**.

- **Servidor OAuth 2.0 externo**: Se delega la autenticación el proveedor de identidades OAuth 2.0 de nuestra elección.

- **Servidor OIDC externo**: La autenticación es delegada en el proveedor de identidades OIDC que se elija.

- **Delegación**: Este mecanismo permite delegar la autenticación en un sitio web para controlar el inicio de sesión.

- **Autenticación mediante entidad administrada**: Este mecanismo consiste en delegar la autenticación en el componente *Azure Active Directory*. En este caso, el sistema *Azure Active Directory* actúa como sistema proveedor de identidades, ya que el cliente realizará una petición a una API determinada, y *Microsoft Azure API Management* solicitará un token de acceso al componente *Azure Active Directory* para que el cliente pueda acceder al recurso solicitado. En lo relativo al servicio de seguridad de autorización, el producto *Azure Active Directory* también se encarga de gestionar los distintos permisos asociados a usuarios individuales o a grupos de usuarios, siguiendo el esquema de autorización RBAC. *Azure Active Directory* posibilita el uso de los siguientes protocolos de autenticación:

    - OAuth 2.0

    - OpenID Connect

    - SAML 2.0

La administración del producto *Microsoft Azure API Management* puede realizarse mediante una interfaz web o mediante una API REST de administración. La autenticación para administrar la plataforma puede realizarse mediante el uso de formulario web de login de la interfaz gráfica o mediante el uso de tokens SAS (Shared Access Signatures) si se desea usar el API REST de administración. Finalmente, en lo relativo a la autorización de la administración de la plataforma, se sigue el esquema de autorización RBAC donde es posible crear usuarios y grupos, a los que se le asignan distintos permisos, en función, del grado de poder del que disponga cada usuario.

### 5.4.2. Cifrado del tráfico y encriptación

*Microsoft Azure API Management* permite encriptar el tráfico [56] [57] de todas las comunicaciones en las que el mismo se encuentre involucrado (ya sea actuando como cliente o como el servidor) mediante el uso del protocolo HTTPS.

En el marco de la encriptación de información sensible o de la información secreta a proteger, *Microsoft Azure API Management* cuenta con una directiva

que permite la administración de secretos mediante la creación de propiedades [58] en formato clave valor. El algoritmo de cifrado simétrico usado por la directiva es 3DES.

### 5.4.3. Alertas y monitorización

Las propiedades de alertas, logs y monitorización se encuentran ubicadas dentro del componente de la suite de Microsoft *Azure Monitor*. *Azure Monitor* recoge distintas métricas relacionadas con las APIs, algunas de las más importantes son:

- Número de solicitudes por API en un rango de tiempo.
- Número de solicitudes por API correctas.
- Número de solicitudes por API incorrectas.
- Número de solicitudes por API con errores de autorización.

*Azure Monitor* cuenta con otra herramienta llamada *registros de diagnóstico* que posibilita el almacenamiento de información de auditoría para cada comunicación en la que una API se vea involucrada. El producto *Microsoft Azure API Management* envía cada hora los registros de actividad que hayan sido activados por el administrador al componente *Azure Monitor.* Alguna de la información que se almacena en estos registros son: método HTTP, url, protocolo, tamaño de respuesta, información del último error conocido, tiempo de respuesta del backend…

En la *Figura 18. Azure API Management dashboard monitorización* se muestra un dashboard de monitorización para una API de ejemplo:

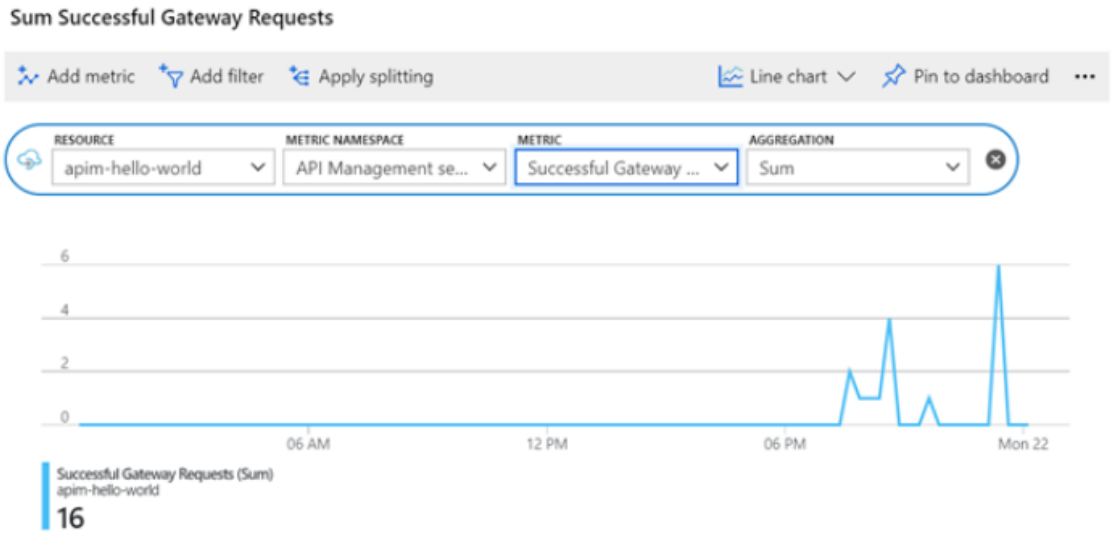



**Figura 18. Azure API Management dashboard monitorización**

**Fuente: https://docs.microsoft.com/es-es/azure/api-management/**

Otra característica del producto *Azure Monitor* es la creación de alertas. Se pueden crear alertas asociadas a métricas o registros de diagnóstico para notificar a los responsables del API de problemas en el menor tiempo posible. *Azure Monitor* admiten distintos canales de notificación cuando una alerta se active:

- Email.
- SMS.
- Web hook.
- Notificación PUSH Azure.
- Llamada telefónica.

### 5.4.4. Protección frente amenazas

Los mecanismos de protección frente amenazas y ataques comunes se delega en el WAF de la suite Microsoft Azure, el producto *Azure Application Gateway*, el cual provee protección a las aplicaciones web situadas en la nube de Microsoft Azure.

En la *Figura 19. Azure API Management protección WAF* vemos un posible escenario, en el que se coloca el WAF *Azure Application Gateway* justamente delante de la aplicación web a proteger, en este caso, el componente *Microsoft Azure API Management*.

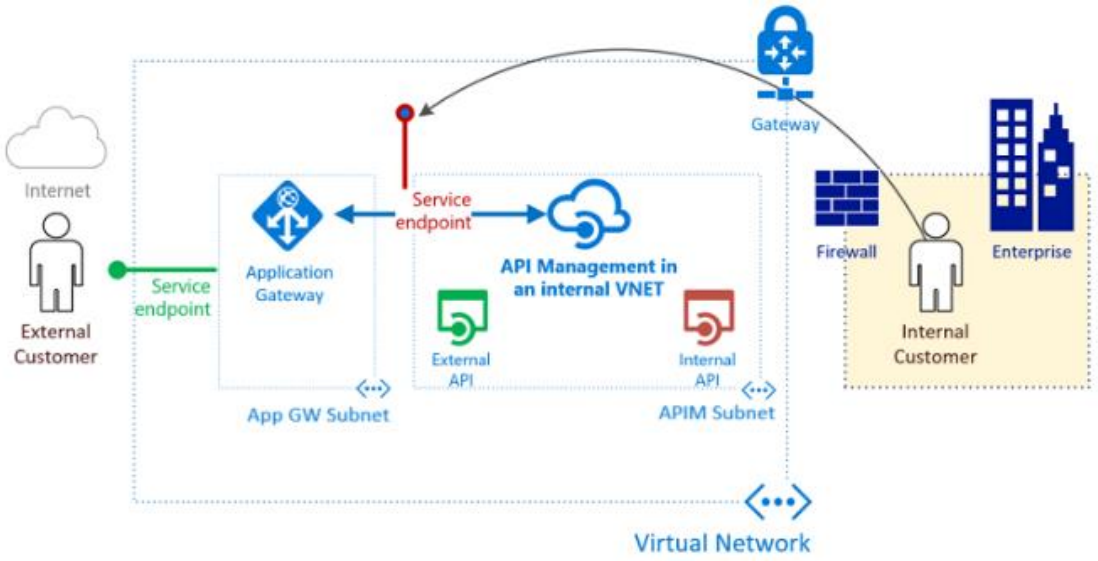



**Figura 19. Azure API Management protección WAF**

**Fuente: https://docs.microsoft.com/es-es/azure/api-management/**

La inclusión del componente *Azure Application Gateway* aumenta la seguridad del sistema de API Management, contando con mecanismos [59] de protección para hacer frente a las siguientes amenazas:

- SQL Injection.
- XSS (Cross Site Scripting).
- Ataques a aplicaciones web comunes: Inyección de comandos, HTTP request smuggling, HTTP response splitting e inclusión de ataques basados en inclusión de archivos remotos.
- Protección frente a violaciones del protocolo HTTP.
- Protección frente bots, crawlers y escáneres.

### 5.4.5. Control y regulación del tráfico

El producto *Microsoft Azure API Management* ofrece distintas posibilidades de limitación del tráfico, en base a distintos parámetros configurables. El sistema de API Management consigue dar esta funcionalidad mediante el uso de distintas directivas de control de tráfico. La utilización de estas directivas resulta un aspecto clave desde el punto de vista de la seguridad, ya que ofrece protección frente a ataques que buscan un funcionamiento inadecuado del sistema de API Management, como por ejemplo los ataques de denegación de servicio. La limitación del tráfico [60] puede realizarse en base a los siguientes aspectos:

- Limitación basada en producto: Se aplican limitaciones en base al tipo de entorno de ejecución. Por ejemplo, mediante esta política se puede limitar el tráfico en el entorno de desarrollo en base a la detección de picos de tráfico.
- Limitación basada en API Key: Se aplica la limitación de tráfico en base a una cuota de servicio definida o en base al número de peticiones en un periodo de tiempo específico.
- Limitación basada en identidad de usuario: Una vez se haya autenticado al usuario que envía la petición al API, se puede limitar la velocidad de consumo para dicho usuario.
- Limitación en base a dirección IP del cliente.

## 5.5. WSO2 API Manager

### 5.5.1. Autenticación y autorización

La solución *WSO2 API Manager* cuenta con los siguientes mecanismos [61] [62] [63] de autenticación y autorización:

- OAuth 2.0: La solución permite el uso del protocolo OAuth 2.0 para realización del proceso de autorización. *WSO2 API Manager* en adición cuenta con un mecanismo para encriptar las claves *OAUth* involucradas en las comunicaciones. *WSO2 API Manager* permite realizar dicha encriptación mediante el uso del algoritmo de criptografía asimétrica RSA o mediante el algoritmo de criptografía simétrica AES.

- XACML: *WSO2 API Manager* admite la posibilidad de brindar el servicio de autorización mediante el protocolo XACML, basado en XML. El producto admite tanto conectarse con un servidor XACML externo como de utilizar el componente de la suite *WSO2 Identity Server* como servidor XACML. En la *Figura 20. Autorización XACML WSO2 API Manager* se muestra el uso del mecanismo de autorización basado en XACML usando el componente *WSO2 Identity Server*:

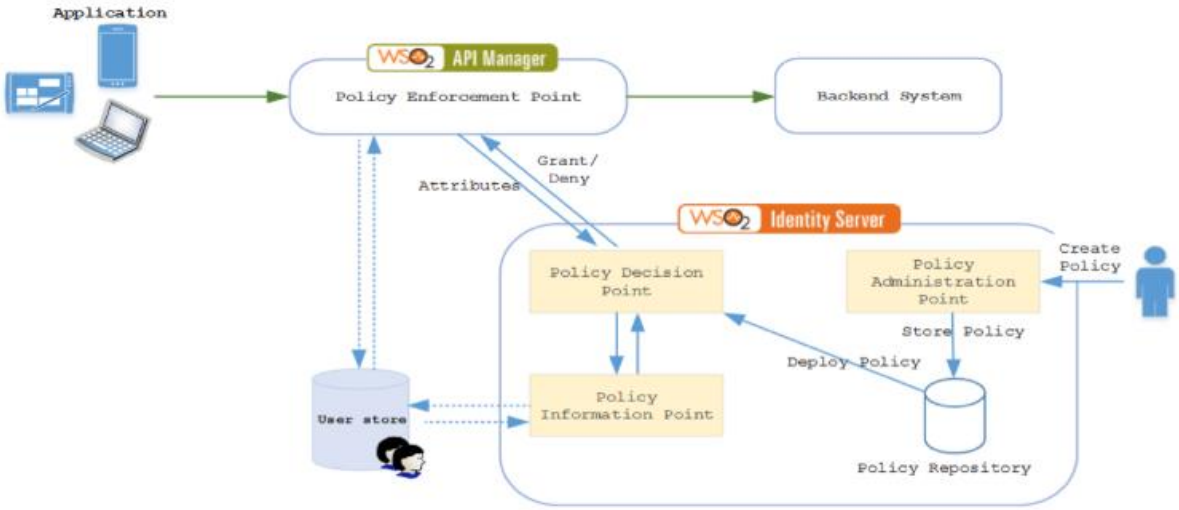



**Figura 20. Autorización XACML WSO2 API Manager**

**Fuente: https://docs.wso2.com/display/AM260/**

- SAML 2.0: Posibilidad de usar el protocolo SAML 2.0 para la federación de identidades en el componente *WSO2 API Manager*. Se acepta en el producto la conexión con cualquier proveedor de identidad SAML externo o usar el componente de la suite de *WSO2 Identity Server* para que actué como proveedor de identidad SAML. En la *Figura 21. SAML WSO2 API Manager* se ilustra el diagrama de comunicaciones llevadas

a cabo para federar una identidad mediante el uso del protocolo SAML en el componente WSO2 API Manager:

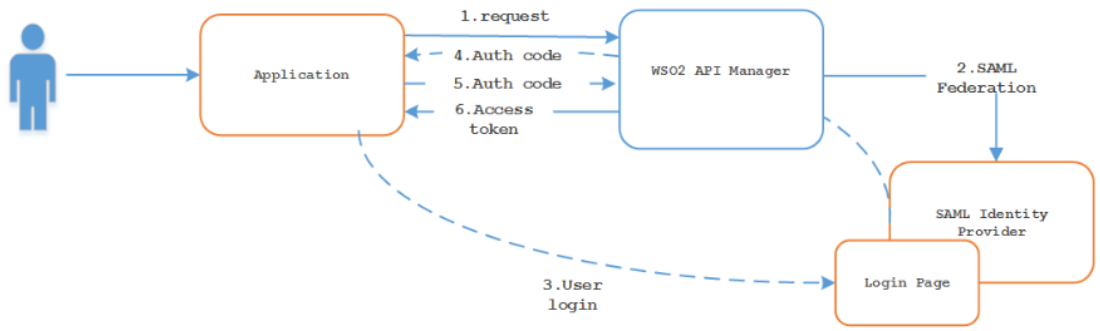



**Figura 21. SAML WSO2 API Manager**

**Fuente: https://wso2.com/library/articles/2017/03/use-cases-of-utilizing-saml-with-wso2-api-manager/**

- Open ID Connect: El protocolo Open ID Connect es un protocolo que proporciona el servicio de seguridad de autenticación y que se sitúa una capa por encima del protocolo OAuth que brinda el servicio de autorización. *WSO2 API Manager* permite brindar un servicio de federación de identidades y de Single Sign On mediante el uso del protocolo Open ID Connect. Para poder implementar este mecanismo es necesario indicar la ubicación del servidor Open ID Connect.

- Autenticación básica.

- Autenticación mutua SSL: Consistente en que ambos extremos de la comunicación puedan validar el certificado de cada miembro involucrado en dicha comunicación.

- JSON Web Token (JWT): Es un protocolo que brinda autenticación, está basado en el uso de estructuras en formato JSON denominadas tokens y compuestas de cabecera, payload y firma. Este protocolo es útil para el intercambio de información de autenticación de manera segura y en casos, en los que el sistema de API Management necesita mandar información de la identidad de los usuarios que consumen las APIs a los sistemas back-ends. En la *Figura 22. Diagrama JWT* se ilustra el funcionamiento del protocolo, en este caso el componente *WSO2 API Manager* actuaría como servidor de aplicación, y se encargaría de validar la identidad del usuario.

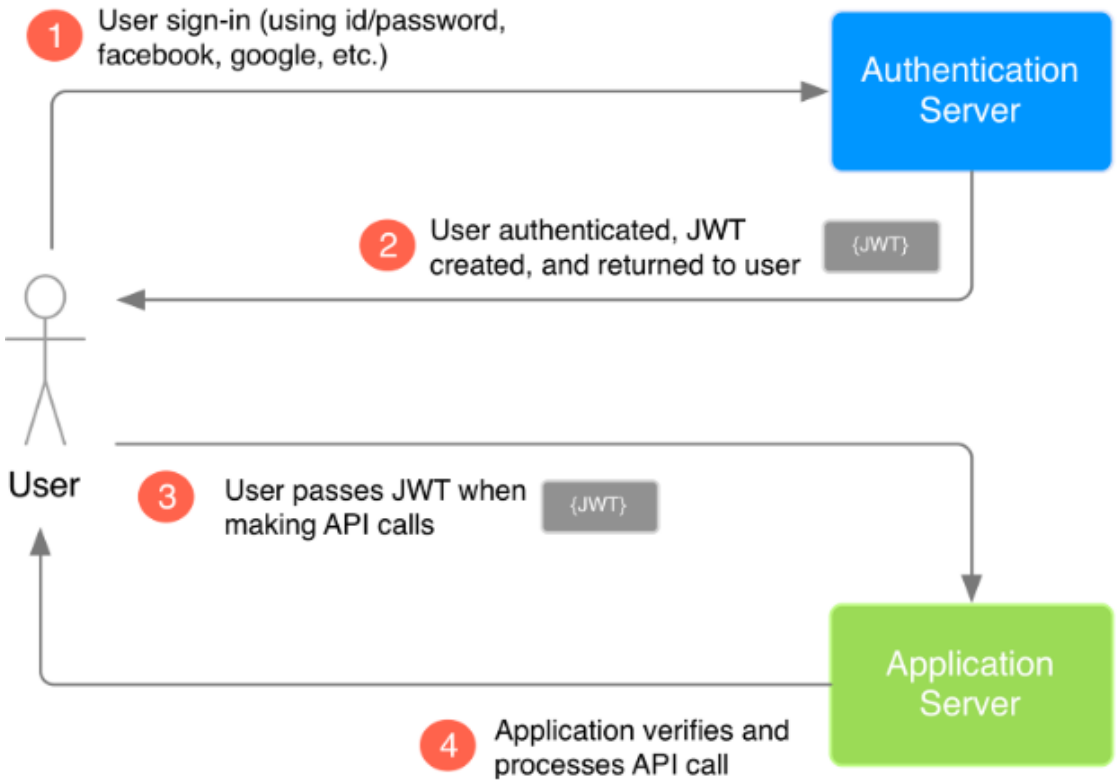



**Figura 22. Diagrama JWT**

**Fuente: https://medium.com/vandium-software/5-easy-steps-to-understanding-json-web-tokens-jwt-1164c0adfcec**

### 5.5.2. Cifrado del tráfico y encriptación

El producto *WSO2 API Manager* cuenta con mecanismos y estrategias para cifrar el tráfico mediante SSL/TLS [64] con el uso del protocolo HTTPS. Es decir, el producto posee capacidades para publicar las APIs mediante el protocolo HTTPS y para comunicarse con los distintos sistemas back-ends mediante el uso de este protocolo. En adición, *WSO2 API Manager* concede la posibilidad de establecer autenticación mutua y que los dos extremos de la comunicación validen sus certificados, tanto en la zona inbound (comunicaciones desde clientes hacia API Management) como en la zona outbound (comunicaciones desde API Management hacia los back-ends).

*WSO2 API Manager* brinda herramientas para realizar el enmascaramiento de la información sensible en los logs del sistema [65], con objeto de que dicha información sensible no quede expuesta.

En lo relativo al almacenamiento y recuperación de información sensible y contraseñas de forma segura, *WSO2 API Manager* cuenta con el mecanismo *Secure Vault* [66], que permite almacenar la información cifrada y asignarle alias para su posterior recuperación mediante el proceso de descifrado. La característica *Secure Vault* ofrece diversos ámbitos de aplicación a la hora de realizar la encriptación, ya que es posible encriptar la información que se encuentre ubicada en contraseñas de archivos de configuración del producto

*WSO2 API Manager* o encriptar propiedades personalizadas definidas a lo largo del flujo del API.

### 5.5.3. Alertas y monitorización

En el ámbito de la monitorización *WSO2 API Manager* cuenta con el componente integrado *API Manager Analytics* que provee de informes, estadísticas y gráficos con información de interés de las distintas APIs desplegadas en el componente API Management. En adición puede configurarse alertas asociadas a determinados aspectos de las APIs para detectar una actividad inusual o administrar las distintas localizaciones geográficas de los clientes en base a analíticas de geolocalización. Algunas de las estadísticas [67] más importantes que pueden consultarse son:

- Invocaciones realizadas a un API en función de parámetros como el usuario y versión.
- Número de APIs creadas en un rango de tiempo.
- Última vez que una API fue invocada.
- Consumo de recursos del sistema por parte de una API.
- Rutas de recursos invocadas por cada API.
- Número de peticiones erróneas por API.
- Latencia asociada a cada API.
- Consumo de recursos del sistema de un API clasificada geográficamente.
- Número de suscripciones por API.
- Número de inicio de sesión de desarrolladores por unidad de tiempo.
- Top usuarios por API.

Otra característica con la que cuenta el producto es que además de proporcionar las métricas y estadísticas anteriores, es capaz de integrarse con el servicio de analíticas de Google: *Google Analytics*.

*WSO2 API Manager* permite crear alertas para notificar diferentes sucesos. Normalmente, las alertas son usadas para informar de la ocurrencia de un fallo repentino en una o varias APIs, como por ejemplo un anormal tiempo de respuesta o el uso de una ruta de recurso inusual dentro de la invocación a una API. *WSO2 API Manager* cuenta con la siguiente tipología de alertas [68] preconfiguradas:

- Tiempo de respuesta anormal.
- Tiempo de respuesta del back-end anormal.
- Número de peticiones anormal.
- Acceso a un recurso con ruta anormal en la invocación a una API.
- Detección de IP sospechosa: Para detectar cambios en la IP de clientes que deben tener una IP fija.
- Rebase del plan de cuota: Esta alerta se lanza para notificar a un cliente que ha alcanzo el número máximo de peticiones establecido para su plan de uso.
- Disponibilidad de las APIs: Esta alerta se activa cuando una API devuelve códigos de respuesta comprendidos entre 500 y 600 o cuando el tiempo de respuesta de la API rebasa un determinado valor preconfigurado.

Finalmente, en lo que respecta a los canales de notificación disponibles para las alertas anteriormente expuestas, el único canal de notificación posible es el de notificación mediante correo electrónico.

### 5.5.4. Protección frente amenazas

La solución *WSO2 API Manager* cuenta con tres mecanismos [69] para hacer frente a distintos ataques a las aplicaciones web. Estos tres mecanismos son políticas de validación que evalúan distintos campos de la petición entrante del usuario al sistema API Management:

- Protección de amenazas mediante expresiones regulares: *WSO2 API Manager* provee de una serie de expresiones regulares para realizar validaciones sobre diversos campos de la petición como el contenido del mensaje, parámetros de consulta, URI, cabeceras HTTP… Si se detecta alguna coincidencia con alguno de los patrones definidos, entonces la petición es bloqueada. Mediante el uso de esta política es posible definir expresiones regulares para hacer frente a los siguientes ataques:
    - SQL Injection.
    - JavaScript Injection.
    - XPath Injection.
    - Java Exception Injection.
    - Server-side Injection.

- XPath Abbreiviated Injection.

- Protección frente ataques de estructura JSON: Esta política valida el cuerpo en formato JSON de la petición entrante para protegerse frente ataques basados en payloads JSON. En la política es posible configurar la siguientes propiedades:

  - maxPropertyCount: Máximo número de campos que puede tener el body del mensaje en formato JSON.

  - maxStringLength: Umbral para especificar el tamaño máximo de cada tipo String del mensaje JSON.

  - maxArrayElementCount: Límite de elementos array en el mensaje JSON.

  - maxKeyLength: Máximo número de elementos de tipo clave en la estructura del mensaje JSON.

  - maxJsonDepth: Máximo de elementos anidados permitidos en la estructura del mensaje JSON.

- Protección frente ataques de estructura XML: Mediante esta política es posible validar el cuerpo en formato XML de la petición entrante para protegerse frente ataques basados en payloads XML. En la política es posible configurar la siguientes propiedades:

  - dtdEnabled: Limitación en el uso de archivos *.dtd* en el mensaje XML.

  - externalEntitiesEnabled: Propiedad para activar o desactivar el uso de entidades XML externas en el cuerpo del mensaje XML de la petición entrante del usuario.

  - maxXMLDepth: Máximo nivel de anidamiento permitido en la estructura XML.

  - maxElementCount: Límite de nodos permitidos en la estructura XML.

  - maxAttributeCount: Máximo número de atributos permitidos en el mensaje XML.

  - maxAttributeLength: Extensión máxima permitida para cada atributo de la estructura XML.

  - maxChildrenPerElement: Máximo número de nodos hijos permitidos por nodo en la estructura XML.

## 5.5.5. Control y regulación del tráfico

Dada la multitud de agentes implicados en el consumo de una API, es necesario el establecimiento de mecanismos de control de tráfico para asegurar el correcto funcionamiento del propio sistema API Management y de los back-ends, de lo contrario ambos podrían llegar a saturarse. En adición, la misión de estos mecanismos de control del tráfico no es únicamente la de no saturar a los distintos servidores, sino que también poseen una labor fundamental en el ámbito de monetización de las APIs para controlar que el plan de uso al que está abonado cada consumidor es llevado realmente a cabo. En la *Figura 23. Control de tráfico WSO2 API Manager* puede verse como es posible asignar distintas límites de tráfico a los distintos agentes implicados en proceso de consumo de las APIs:

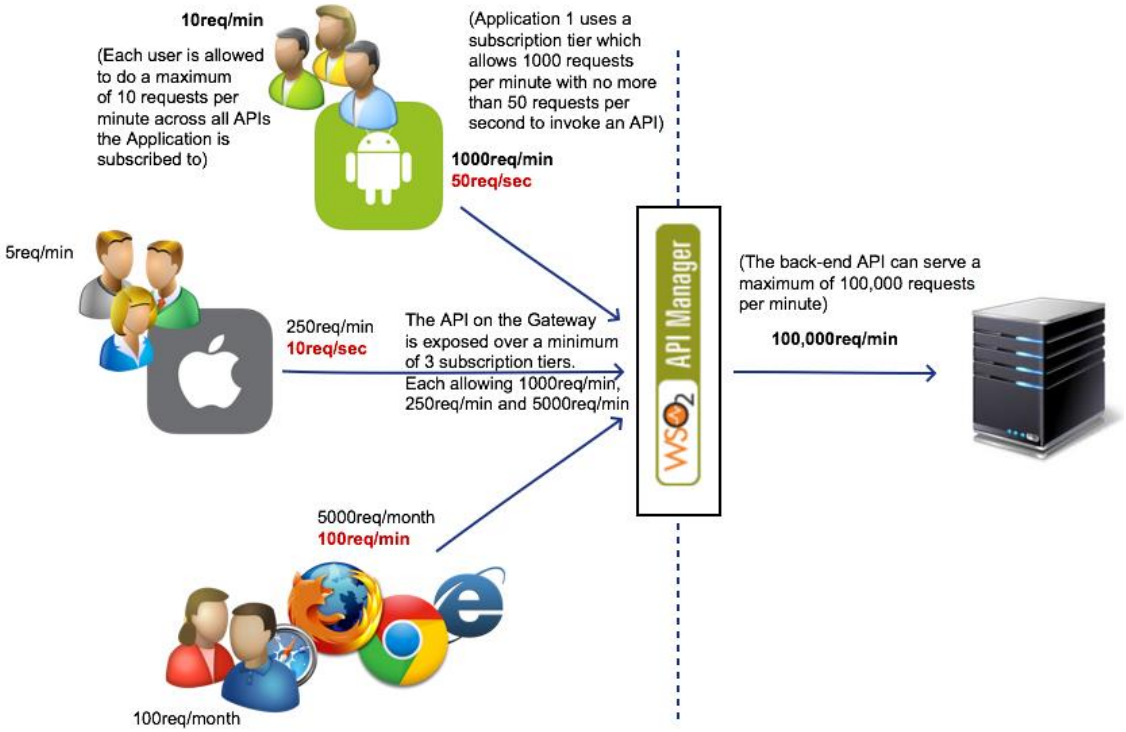



**Figura 23. Control de tráfico WSO2 API Manager**

**Fuente: https://docs.wso2.com/display/AM260/Introducing+Throttling+Use-Cases**

*WSO2 API Manager* permite establecer controles [70] para realizar las siguientes limitaciones del tráfico:

- Máximo Caudal para el back-end: Establece el número máximo de peticiones por segundo que una API puede enviar a un back-end.

- Limitación en base al plan de uso: Se limita el número máximo de peticiones por minuto en base al plan al que esté subscrito el usuario,

estando disponibles los siguientes planes:

  - Gold: 5000 peticiones por minuto.
  - Silver: 2000 peticiones por minuto.
  - Bronze: 1000 peticiones por minuto.

- Políticas de limitación avanzadas: Estas políticas permite establecer mecanismos de control de tráfico en base a distintos aspectos como:
  - Limitación por dirección IP específica o por rango de direcciones IP: Es posible definir cuotas de uso para una dirección IP o para un rango de direcciones específico.
  - Limitación en base a cabeceras HTTP: Esta estrategia resulta de utilidad cuando queremos restringir el número de peticiones en base a parámetros de cabeceras HTTP, como por ejemplo el tipo de contenido.
  - Limitación en base a parámetros JWT.
  - Limitación en base a parámetros de consulta de la petición HTTP: Es posible establecer unas condiciones de tráfico para cada parámetro de consulta.

# 6. Síntesis y comparativa

En este capítulo se realizará una comparativa de las distintas soluciones de sistemas API Management estudiadas en el presente trabajo de fin de máster atendiendo a criterios como mecanismos para combatir las amenazas OWASP más importantes, riqueza en los servicios de seguridad estudiados en el capítulo *5. Análisis de servicios de seguridad* de esta memoria y otros aspectos como conciencia de seguridad y educación al usuario.

En primer lugar realizaremos una codificación de las diez amenazas de las aplicaciones web según OWASP, para poder hacer referencia a ellas de una manera más cómoda. Se define la siguiente codificación:

| Amenaza | Código de Amenaza |
|---|---|
| **Inyección** | **A1** |
| **Pérdida de autenticación** | **A2** |
| **Exposición de datos sensibles** | **A3** |
| **Entidades externas XML** | **A4** |
| **Pérdida de control de acceso** | **A5** |
| **Configuración de seguridad incorrecta** | **A6** |
| **Secuencias de comandos en sitios cruzados (XSS)** | **A7** |
| **Deserialización insegura** | **A8** |
| **Componentes con vulnerabilidades conocidas** | **A9** |
| **Registro y monitorización insuficientes** | **A10** |

**Tabla 3. Codificación vulnerabilidades OWASP**

La existencia de mecanismos para combatir las vulnerabilidades anteriores resulta fundamental a la hora de evaluar la robustez de un sistema desde el punto de vista de la seguridad. A continuación veremos cómo cada uno de los sistemas API Management estudiados hacen frente a dichas vulnerabilidades. La realización de este sistema de evaluación permite de una manera rápida y sencilla obtener una visión general del nivel de blindaje y seguridad que posee cada solución API Management estudiada en lo relativo a la protección frente a las principales amenazas actuales de las aplicaciones web.

En las tablas ulteriores se plasman los mecanismos de los sistemas API Management estudiados [71] [72] [73] [74] para dar solución a las principales amenazas de las aplicaciones web anteriormente expuestas:

| Apigee | |
|---|---|
| **Vulnerabilidades** | **Mecanismos contra amenazas** |
| **A1** | • Política *JSONThreatProtection*.<br>• Política *XMLThreatProtetction*.<br>• Política *RegularExpressionProtection*.<br>• Validaciones de parámetros y cabeceras de la petición mediante el uso de la política JavaScript. |
| **A2** | • Validación del API Key.<br>• OAuth 2.0.<br>• JWT. |
| **A3** | • Soporte TLS/SSL en todas las comunicaciones en las que Apigee se encuentra involucrado.<br>• Soporte para 2-way TLS.<br>• Mecanismo *data masking* para ocultar información sensible.<br>• Mecanismo key-value maps. |
| **A4** | • Política *XMLThreatProtetction*. |
| **A5** | • Soporte para SSO con proveedor de identidad externo.<br>• Esquema de autorización RBAC.<br>• Mecanismo *data masking* para ocultar información sensible.<br>• Mecanismo key-value maps. |
| **A6** | • Uso de flujos compartidos.<br>• Apigee garantiza la protección frente a vulnerabilidades en librerías de terceros.<br>• Sistema de notificación de nuevas vulnerabilidades y actualización automática de parches (versión cloud) o mediante notificación (versión on-premises) |
| **A7** | • Política *RegularExpressionProtection*.<br>• Soporte CORS mediante política *AssignMessage*. |
| **A8** | • Política *JSONThreatProtection*.<br>• Política *XMLThreatProtetction*.<br>• Política *RegularExpressionProtection*.<br>• Política JavaScript.<br>• Protección por el fabricante Apigee a nivel de infraestructura frente a ataques de deserialización.<br>• Política de cache para proteger frente a ataques de deserialización que usan la repetición de mensajes. |
| **A9** | • Sistema de notificación de nuevas vulnerabilidades y actualización automática de parches (versión cloud) o mediante notificación (versión on-premises) |
| **A10** | • Disponibilidad de analíticas a nivel de API.<br>• Monitorización de salud de las APIs.<br>• Interfaz gráfica de monitorización de APIs y dashboards asociados.<br>• Mecanismos de captura de fallos en APIs.<br>• Logs a nivel de API, productos y organización.<br>• Definición de alertas. |

**Tabla 4. Mecanismos OWASP Apigee**

| CA API Management | |
|---|---|
| **Vulnerabilidades** | **Mecanismos contra amenazas** |
| **A1** | • Provisión de políticas de validación para protegerse frente ataques de inyección SQL y frente a otro tipo de inyecciones.<br>• Mecanismos de acceso tanto al contenido como al contexto de la solicitud y de la respuesta HTTP. Estos mecanismos permiten la inspección y protección en tiempo real. |
| **A2** | • Oferta de mecanismos de autenticación robustos y de mecanismos de autenticación multi-factor.<br>• Protección contra ataques de fuerza fruta que buscan romper contraseñas mediante iteraciones de prueba y error.<br>• Mecanismo de protección frente ataques basados en sesiones. Mediante el control de atributos de las cookies de seguridad, usando técnicas de firmado y cifrado o mediante el uso de identificadores para asegurar sticky sessions. |
| **A3** | • Mediante la gestión de diversas políticas es posible realizar un cifrado de los datos estáticos y de los datos en tránsito, pudiéndose llegar a configurar dichas políticas para asegurar la integridad y lograr el nivel de exposición deseado de los datos sensibles, en función de normas como PCI-DSS, para que la información cumpla con los estándares exigidos en industrias como la sanidad, el sector financiero o el sector público. |
| **A4** | • Disposición de mecanismos para la protección frente ataques basado en entidades XML externas para protegerse contra la ejecución remota de código y ataques de denegación de servicio, mediante el uso de distintas políticas. |
| **A5** | • *CA API Management* ofrece una amplia gama de mecanismos de control de acceso, tanto basados en soluciones propietarias como en estándares de la industria, para garantizar que los usuarios y las aplicaciones puedan acceder a los recursos protegidos con el nivel de privilegios necesarios y mediante el uso de políticas de seguridad centralizadas. |
| **A6** | • *CA API Management* cuenta con un API Gateway en el que la seguridad ha sido tenida en cuenta desde el principio y donde esta cobra un propósito de especial importancia. El componente API Gateway ha sido reforzado mediante distintas técnicas de bastionado para que su instalación sea fácil y segura en una zona DMZ. CA acredita lo expuesto anteriormente cumpliendo con la certificación *Common Criteria* en distintos ámbitos de seguridad. |
| **A7** | • *CA API Management* además de contar con políticas para hacer frente a ataques de tipo *Cross Site Scripting*, es una solución reforzada y diseñada específicamente para proteger frente a ataques que se realicen a servicios, API y aplicaciones, posibilitando a los clientes la detección, respuesta y bloqueo frente a dichos ataques, utilizando una política de seguridad centralizada, como si se tratará de un firewall de la capa de aplicación. |
| **A8** | • *CA API Management* cuenta con políticas para hacer frente a |

| | |
|---|---|
| | los ataques que tienen lugar en el proceso de deserialización. Estos mecanismos se basan en la inspección y establecimientos de limitaciones en la deserialización de los mensajes en formato JSON o XML. |
| **A9** | • En adición a lo comentado en la fila *A5* de esta tabla, donde se reflejaba que *CA API Gateway* es un sistema API Gateway especializado en la seguridad y que cumple con la especificaciones de seguridad *Common Criteria*, CA posee un equipo de ingenieros que está continuamente vigilando la aparición de nuevas vulnerabilidades, para estudiarlas, notificar y liberar rápidamente parches de seguridad que eliminen dichas vulnerabilidades de los productos de sus clientes. Estos parches de seguridad pueden aplicarse de una manera fácil con el sistema de gestión de parches que se encuentra incluido en el componente *CA API Management*. |
| **A10** | • Capacidad para implementar niveles definibles y personalizados de monitorización, que posibiliten el apropiado nivel de notificación ante eventos e incidentes, en función de las necesidades de cada organización |

**Tabla 5. Mecanismos OWASP CA API Management**

| **Amazon API Gateway** | |
|---|---|
| **Vulnerabilidades** | **Mecanismos contra amenazas** |
| **A1** | • Componente *AWS WAF*: Este componente tiene capacidades para evitar y mitigar ataques de inyección SQL. Es posible el uso de distintas condiciones de coincidencia de inyección SQL para implementar reglas que eviten dichos intentos inyección. |
| **A2** | • Componente *AWS WAF*: Mecanismo de blacklist para peticiones procedentes de clientes, cuando se detecten ataques relativos a secuestros de tokens.<br>• Componentes *AWS IAM*, *AWS Cognito* y *AWS IAM*: Estos componentes oferta distintos mecanismos basados en protocolos robustos de autenticación. |
| **A3** | • Componente Amazon CloudFront: Permite proteger la información sensible en tránsito y la de carácter estático, mediante el uso de algoritmos de cifrado. |
| **A4** | Sin mecanismos. |
| **A5** | • Componente *Amazon IAM*: Ofrece mediante sus políticas de recursos un mecanismo de control robusto, basado en el modelo de autorización RBAC. |
| **A6** | • Componente *Amazon Inspector*: Escáner de seguridad que realiza una evaluación de la configuración de seguridad de la aplicación y busca posibles vulnerabilidades.<br>• En cuanto al nivel de actualización frente a incidencias y aplicación de parches de seguridad es totalmente transparente para el usuario, ya que al ser una solución puramente Cloud, es el equipo de Amazon el que realiza esta actuación por el usuario y se compromete con sus clientes a brindar una |

| | |
|---|---|
| | actuación rápida y de calidad. |
| **A7** | • Componente *AWS WAF*: Este firewall a nivel de aplicación contiene una serie de políticas que comprueban la ocurrencia de intentos de ataques basados en XSS mediante el análisis de distintos campos de la petición web como el mensaje, cookies, URL y parámetros de la petición. |
| **A8** | Sin mecanismos |
| **A9** | • Componente *Amazon Inspector*: Escáner que realiza una evaluación de la configuración de seguridad de la aplicación y busca posibles vulnerabilidades.<br>• Múltiples escáner de vulnerabilidades disponibles en *AWS MarketPlace* como por ejemplo el conocido escáner Rapid7. |
| **A10** | • Componente Amazon X-Ray: Monitorización de peticiones y seguimiento de trazas.<br>• Componente Amazon CloudWatch: Generación y visualización de métricas y definición de alertas y canales de notificación.<br>• Componente Amazon CloudTrail: Monitorización de métricas relativas a la administración del sistema API Management. |

**Tabla 6. Mecanismos OWASP Amazon API Gateway**

| **Microsoft Azure API Management** | |
|---|---|
| **Vulnerabilidades** | **Mecanismos contra amenazas** |
| **A1** | • Componente *Azure WAF*: El firewall a nivel de aplicación de la suite Azure cuenta con mecanismos para hacer frente a ataques basados en inyección SQL.<br>• Componente *Mod Security*: Es un WAF open source que se instala como un módulo adicional en las aplicaciones de la suite de Azure permitiendo realizar un exhaustivo filtrado del tráfico entrante a la aplicación.<br>• Componente *Azure Resource Manager*. |
| **A2** | • Uso de protocolos como *SAML*, *OAuth*, *WS Federation*.<br>• Compatibilidad con autenticación multi factor.<br>• Componente *Azure Active Directory.* |
| **A3** | • Mecanismos de encriptación de base de datos.<br>• Enmascaramiento de datos dinámico.<br>• Mecanismo Azure Key Vault.<br>• Cifrado de transporte SSL. |
| **A4** | Sin mecanismos. |
| **A5** | • Autorización delegada en grupos de Azure Active Directory.<br>• Mecanismo de autenticación basado en notificaciones. |
| **A6** | • Modelo RBAC para controlar los privilegios de las distintas aplicaciones.<br>• Componente Azure Security Center: Realiza una evaluación en busca de configuraciones débiles de seguridad en los productos de la suite Azure.<br>• El componente Azure WAF consta de mecanismos base para detectar errores comunes en un número acotado de productos como por ejemplo, Apache y Microsoft IIS. |

| | |
|---|---|
| **A7** | • El componente Azure WAF posee en su configuración base una serie de reglas de filtrado para hacer frente a ataques de tipo Cross Site Scripting. |
| **A8** | Sin mecanismos. |
| **A9** | • Escáner de vulnerabilidades *Tinfoil*: Ayuda a detectar la existencia de componentes vulnerables en las aplicaciones web de la suite Azure y en los componentes de las mismas. |
| **A10** | • Componente *Azure Monitor* para la definición y generación de métricas y alertas de las distintas APIs. |

**Tabla 7. Mecanismos OWASP Microsoft Azure API Management**

| **WSO2 API Manager** | |
|---|---|
| **Vulnerabilidades** | **Mecanismos contra amenazas** |
| **A1** | • Política de detección basada en expresiones regulares para combatir distintos tipos de ataques basados en inyecciones como SQL, Server-Side, XPath… |
| **A2** | • Uso de estándares de la industria como OAuth, XACML, SAML… |
| **A3** | • Mecanismo *Secure Vault* para el cifrado de la información sensible.<br>• Herramientas para el enmascaramiento de información crítica. |
| **A4** | • Política de protección que permite desactivar el uso de entidades externas XML bajo demanda. |
| **A5** | • Soporte para SSO con proveedor de identidad externo.<br>• Esquema de autorización RBAC. |
| **A6** | Sin mecanismos. |
| **A7** | • WSO2 cuenta con un mecanismo de protección frente a ataques de tipo Cross Site Scripting, el cual puede ser activado en función de las distintas rutas web que se le especifiquen en la configuración. |
| **A8** | Sin mecanismos. |
| **A9** | Sin mecanismos. |
| **A10** | • Módulo de integración con Google Analytics.<br>• Componente *API Manager Analytics* integrado en el propio sistema API Management con capacidades para definir niveles de alertas, notificaciones y generar métricas configurables. |

**Tabla 8. Mecanismos WSO2 API Manager**

Para establecer la calificación final de las distintas soluciones de API Management estudiadas en el presente trabajo de fin de máster se han tenido en cuenta diversos criterios y aspectos de seguridad:

- **Autenticación y autorización**: Variedad y robustez de mecanismos de autenticación y autorización ofertados por la solución API Management objeto de estudio.

- **Cifrado del tráfico y encriptación**: Valoración del cifrado y la encriptación de las distintas soluciones API Management, en función del volumen de mecanismos y la calidad de los mismos.

- **Alertas y monitorización**: Evaluación de la diversidad de métricas y alertas permitidas por las soluciones API Management, así como el número de canales de notificación disponibles y el nivel de configuración permitido de las funciones de monitorización.

- **Protección frente amenazas**: Valoración del número y robustez de mecanismos predefinidos en las soluciones API Management para hacer frente a amenazas y ataques.

- **Control y regulación del tráfico**: Evaluación de las herramientas para el cifrado y enmascaramiento de información sensible y mecanismos para el cifrado del tráfico.

- **Mecanismos de protección frente a las amenazas OWASP TOP 10**: Calificación en función de la cobertura que poseen las soluciones API Management frente a las top diez vulnerabilidades definidas por OWASP.

- **Centralización y funcionamiento de los aspectos de seguridad**: Se valorará de manera positiva que las funciones de seguridad se encuentre auto contenidas en un mismo producto y que los aspectos de seguridad no se encuentren difuminados en una multitud de componentes o productos aislados, ya que este enfoque disgregativo aunque promueve la interacción de componentes y no comporta un único punto de fallo, aumenta la dificultad de la gestión de la seguridad, y el administrador del sistema puede olvidar funcionalidades de seguridad o incluso no darse de cuenta de vulnerabilidades o configuraciones inadecuadas, como producto de la incomodidad de tener las distintas funcionalidades de seguridad separadas del sistema de API Management.

- **Documentación y consciencia de seguridad del fabricante**: Dado que la seguridad tiene que ser tenida en cuenta desde un principio y ser presentada de una manera amistosa al usuario, sin que parezca que el administrador por el hecho de hacer su sistema seguro tendrá que pagar el coste de un elevado conocimiento técnico, se ha calificado de manera positiva el hecho de que las distintas funciones de seguridad de las soluciones API Management se hayan tenido presentes durante toda la documentación en la medida de lo posible. En adición se ha evaluado de manera favorable la aparición de guías de seguridad y buenas prácticas en la configuración de las soluciones API Management estudiadas, como la existencia de vídeos relativos a diferentes aspectos de seguridad y de concienciación a los usuarios sobre los riesgos que comporta una mala administración de la misma.

A continuación se realizará una codificación de los aspectos de seguridad contemplados en la evaluación de los distintos sistemas de API Management para aumentar la flexibilidad a la hora de evaluarlos matemáticamente y se le establecerá un peso de calificación:

| Amenaza | Código de Amenaza | Peso de calificación |
|---|---|---|
| **Autenticación y autorización** | E1 | **1.2** |
| **Cifrado del tráfico y encriptación** | E2 | **1.2** |
| **Alertas y monitorización** | E3 | **1.2** |
| **Protección frente amenazas** | E4 | **1.2** |
| **Control y regulación del tráfico** | E5 | **1.2** |
| **Mecanismos de protección frente a las amenazas OWASP TOP 10** | E6 | **2** |
| **Centralización y funcionamiento de los aspectos de seguridad** | E7 | **1.5** |
| **Documentación y consciencia de seguridad del fabricante** | E8 | **0.5** |

**Tabla 9. Codificación aspectos de seguridad**

Por tanto la nota final de cada solución API Management estudiada será el resultado de evaluar la siguiente fórmula matemática:

$$Calificacion = E1*0.12 + E2*0.12 + E3*0.12 + E4*0.12 + E5*0.12 + E6*0.2 + E7*0.15 + E8*0.05$$

En primer lugar se expondrá la calificación por separado de cada solución API Management objeto de estudio en el presente trabajo de fin de máster, y finalmente se presentará una tabla comparativa con la calificación final de cada una de las soluciones en base a los criterios expuestos anteriormente.

| Apigee | |
|---|---|
| **Aspectos Evaluados** | **Calificación** |
| **E1** | 7 |
| **E2** | 9 |
| **E3** | 9 |
| **E4** | 6 |
| **E5** | 7 |
| **E6** | 9 |
| **E7** | 9 |
| **E8** | 10 |

**Tabla 10. Evaluación final Apigee**

$$Calificacion = 7*0.12 + 9*0.12 + 9*0.12 + 6*0.12 + 7*0.12 + 9*0.2 + 9*0.15 + 10*0.05$$
$$Calificacion = 8.21$$

| CA API Management | |
|---|---|
| **Aspectos Evaluados** | **Calificación** |
| **E1** | 10 |
| **E2** | 10 |
| **E3** | 8 |
| **E4** | 10 |
| **E5** | 8 |
| **E6** | 10 |
| **E7** | 9 |
| **E8** | 7 |

**Tabla 11. Evaluación final CA API Management**

$$Calificacion = 10*0.12+10*0.12+8*0.12+10*0.12+8*0.12+10*0.2+9*0.15+7*0.05$$
$$Calificacion = 9.22$$

| Amazon API Gateway | |
|---|---|
| **Aspectos Evaluados** | **Calificación** |
| **E1** | 9 |
| **E2** | 8 |
| **E3** | 10 |
| **E4** | 8 |
| **E5** | 7 |
| **E6** | 8 |
| **E7** | 4 |
| **E8** | 9 |

**Tabla 12. Evaluación final Amazon API Gateway**

$$Calificacion = 9*0.12+8*0.12+10*0.12+8*0.12+7*0.12+8*0.2+4*0.15+9*0.05$$
$$Calificacion = 7.68$$

| Microsoft Azure API Management | |
|---|---|
| **Aspectos Evaluados** | **Calificación** |
| E1 | 8 |
| E2 | 8 |
| E3 | 9 |
| E4 | 8 |
| E5 | 8 |
| E6 | 8 |
| E7 | 4 |
| E8 | 6 |

**Tabla 13. Evaluación final Microsoft Azure API Management**

$$Calificacion = 8*0.12+8*0.12+9*0.12+8*0.12+8*0.12+8*0.2+4*0.15+6*0.05$$
$$Calificacion = 7.42$$

| WSO2 API Manager | |
|---|---|
| **Aspectos Evaluados** | **Calificación** |
| **E1** | 8 |
| **E2** | 8 |
| **E3** | 7 |
| **E4** | 7 |
| **E5** | 7 |
| **E6** | 7 |
| **E7** | 10 |
| **E8** | 5 |

**Tabla 14. Evaluación final WSO2 API Manager**

$$Calificacion = 8*0.12+8*0.12+7*0.12+7*0.12+7*0.12+7*0.2+10*0.15+5*0.05$$

$$Calificacion = 7.59$$

Finalmente tras los resultados de seguridad obtenidos individualmente para cada una de las soluciones estudiadas en el presente trabajo de fin de máster, a continuación se presenta a modo de comparativa final la siguiente tabla donde puede realizarse una comparación desde el punto de vista de la seguridad de los distintos sistemas de API Management, en función de su respectiva calificación:

| Calificaciones finales | |
|---|---|
| **Solución API Management** | **Calificación final** |
| CA API Management | **9.22** |
| Apigee | **8.21** |
| Amazon API Gateway | **7.68** |
| WSO2 API Manager | **7.59** |
| Microsoft Azure API Management | **7.42** |

**Tabla 15. Comparativa y calificaciones finales**

# 7. Conclusiones y líneas futuras

En la actual realidad tecnológica en la que nos encontramos inmersos existen multitud de servicios de naturalezas muy diversas como servicios móviles, cloud, on-premises… Esta gran diversidad de servicios junto con la aparición de paradigmas como Big Data o IoT ha hecho que surjan diferentes necesidades de integración entre estos servicios para posibilitar un intercambio de datos entre las distintas aplicaciones que se alojan en Internet. Las APIs han jugado un papel clave en este proceso de compartir la información, ya que ellas han sido el mecanismo de mayor aceptación que da solución al problema de esta interacción entre los distintos servicios de la red de redes que es Internet.

Como resultado de este proceso de apificación de los distintos servicios de las aplicaciones, ha surgido una economía alrededor de la API, que persigue la monetización de las mismas. Mediante el uso de determinadas herramientas y mecanismos, los propietarios de las APIs pueden realizar distintas funciones de tarificación a sus clientes en base a distintos aspectos como volumetría o contratación de planes de uso.

Dado el gran volumen de APIs que existen en la actualidad y el aumento que se prevé del número de las mismas, nacen los sistemas de API Management para dotar de orden y de una administración rápida, intuitiva y centralizada al gran número de APIs existentes en una organización. El papel que poseen los sistemas API Management en la actualidad, es crucial desde el punto de vista tanto de la calidad del servicio de administración como el de la seguridad.

La mayoría de los sistemas de API Management representa un único punto de fallo desde el punto de vista de la seguridad, ya que en la mayoría de soluciones empresariales las diversas funcionalidades de administración y de seguridad se encuentran alojadas en un mismo producto, para facilitar la administración del propio ciclo de vida de cada API. Por estos motivos, resulta crucial realizar un buen estudio de la seguridad del sistema API Management a elegir en la organización, así como elaborar una serie de medidas de protección adicionales y la realización de planes de reacción en caso de que la seguridad quede comprometida.

Tras el análisis de cinco de las soluciones de API Management más relevantes del mercado tecnológico actual, se determina que actualmente existen dos grandes filosofías o vertientes en el mundo de los sistemas API Management. La primera de ellas enfocada a la especialización de soluciones de API Management en la que la funcionalidad queda auto contenida en único producto y por otro lado, soluciones de API Management ubicadas dentro de grandes plataformas o suite de productos como la suite AWS de Amazon o la suite Azure de Microsoft, estas plataformas no son plataformas exclusivas de interoperabilidad sino que proporcionan diferentes servicios al usuario sin enfocarse en un nicho especifico, en este tipo de soluciones la construcción de un sistema de API Management robusto se basa en utilizar distintos componentes de la suite para encajar las distintas piezas del puzle y alcanzar la funcionalidad deseada en nuestro sistema de API Management. El carácter

descentralizado, debido al reparto de la funcionalidad en distintos componentes de la suite, puede hacer que el uso de este tipo de soluciones no sean las más adecuadas desde el punto de vista de seguridad para la correcta administración de un sistema de API Management, ya que un administrador puede olvidarse con facilidad de en qué componente se encuentra cada servicio de seguridad y tampoco tendría una visión general de toda la seguridad de su sistema.

En lo que respecta al estudio de los servicios de seguridad realizado en el presente trabajo de fin de máster de las cinco soluciones de API Management presentadas se concluye que:

- Todas las soluciones estudiadas brindan servicios de seguridad notables con una calidad media-alta y adaptándose a los estándares y protocolos de seguridad más robustos y representativos del mercado.

- En las soluciones basadas en suites cloud, la mayoría de los servicios de seguridad no se encuentran embebidos en el propio componente de API Management, sino en distintos componentes de la suite. Este hecho hace que en estas soluciones de API Management los mecanismos de prevención contra amenazas se deleguen en el componente que actúa como firewall de aplicación dentro de la suite.

- Dentro de las soluciones estudiadas, *CA API Management* es el más robusto en el ámbito de la seguridad. Durante su análisis se ha podido discernir de una manera clara, que en dicho sistema API Management la seguridad se ha tenido en cuenta desde sus inicios y construcción, llegando a ser una solución de API Management específica para fines de seguridad, debido a que es la única solución que posee mecanismos para varias tipologías de aplicaciones, como es el caso de la aplicaciones móviles, o es la solución que más mecanismos de seguridad posee en ámbitos muy diversos como el de la autenticación, autorización y la prevención contra amenazas.

- En general, en las soluciones estudiadas existe un visible esfuerzo por parte de los distintos fabricantes en el proceso de concienciación y educación al usuario sobre la importancia de la seguridad, aclarando de manera insistente distintos aspectos de seguridad a lo largo de la documentación o mediante vídeos y manuales de medidas de protección y buenas prácticas.

En lo que respecta a las líneas de trabajo futuras existe un gran abanico de posibilidades para poder continuar este trabajo o apoyarse en él para crear líneas nuevas. Debido al crecimiento de las APIs en la actualidad y a la gran apuesta actual que se está haciendo por el uso de sistemas de API Management este trabajo puede servir como base para abordar alguna de las siguientes líneas futuras:

- Estudio y ejecución de ataques de seguridad a algunos de los sistemas de API Management estudiados.

- Análisis de las futuras vulnerabilidades Top 10 OWASP, realizando una evaluación para verificar que las soluciones de API Management estudiadas cuenta con nuevos mecanismos para prevenirlas y que los fabricantes han sido capaces de adaptarse a estas nuevas vulnerabilidades incorporando los mecanismos pertinentes en sus respectivas soluciones.

- Comparativa de las nuevas soluciones API Management informadas en el futuro informe de la consultora *Gartner* con las presentadas en este trabajo de fin máster, con objeto de realizar un examen de la evolución de la seguridad en estos sistemas y determinar qué fabricantes han empeorado o mejorado en el ámbito de la seguridad.

- Desarrollo teórico y práctico de un sistema que asegure la transparencia e integridad a los clientes en el proceso de tarificación llevado a cabo por los propietarios de las APIs, mediante el estudio de tecnologías que permitan brindar esa transparencia y confianza, como por ejemplo tecnologías blockchain.

# 8. Glosario

**API REST:** Conjunto de reglas y especificaciones usados por las distintas aplicaciones para poder intercambiar información y comunicarse entre sí, mediante el uso del protocolo HTTP.

**API Key:** Cadenas alfanuméricas o numéricas que se expiden a los consumidores de las APIs para autenticarlos o para realizar funciones de tarificación.

**URL (Universal Resource Locator):** Identifica una dirección específica que está asociada a una página web con objeto de facilitar a los usuarios su fácil identificación.

**API Mock:** Es una API que se utiliza para funciones de simulación y testeo, y que posee únicamente datos de prueba.

**Comunicación Inbound:** Comunicación realizada desde el cliente al sistema de API Management.

**Comunicación Outbound:** Comunicación realizada desde el sistema de API Management hacia el sistema back-end.

**SLAs:** Los SLAs o *Service Level Agreentments* son los distintos compromisos acordados entre un proveedor de servicios y su cliente, donde se fija unos límites en determinadas funcionalidades del servicio prestado y unos compromisos de calidad.

**Endpoint:** Se denomina endpoint a la URL asociada a la API de un servicio.

**Back-end:** Es la parte de un servicio donde reside la lógica del negocio y que no es directamente accesible por los usuarios. En el contexto de este trabajo de fin de master, el sistema de API Management es el encargado de derivar las distintas peticiones realizadas por los usuarios a su correspondiente backend asociado a cada API.

**SDK:** Conjunto de herramientas que permiten la programación de aplicaciones.

**HTTP:** Protocolo usado en las comunicaciones web, en este protocolo se definen las especificaciones y bases de la arquitectura web. Es un protocolo sin estado.

**HTTPS:** Versión segura del protocolo HTTP.

**Payload:** Carga útil de datos del conjunto de datos transmitidos enviados en la comunicación.

**XML:** Estándar cuya finalidad es la proveer una estructura para la representación de información, mediante el uso de un lenguaje de etiquetas, nodos y anidamiento.

**JSON:** Estándar cuya finalidad es la proveer una estructura para la representación de información. Está compuesto de dos estructuras: una colección de pares clave-valor y una lista ordenada de valores.

**OWASP:** Comunidad abierta dedicada a habilitar a las organizaciones para desarrollar, comprar y mantener aplicaciones confiables.

**Serialización:** Es el proceso por el cual se pueden convertir determinados objetos pertenecientes a una aplicación a un flujo de bytes para ser almacenados en distintos medios de almacenamiento o enviados a través de la red.

**Deserialización:** Proceso inverso a la serialización.

**Logs:** Archivo de texto en el que se registran de manera cronológica los distintos eventos que han tenido lugar dentro de un sistema informático.

**Dashboard:** Representación gráfica de los principales indicadores claves para el negocio, cuya principal orientación es la toma de decisiones en procesos que beneficien a la organización.

**WAF:** Un WAF o Web Application Firewall es un cortafuegos específico para aplicaciones web. La mayoría de WAF se utilizan de la misma manera: suponen un muro de defensa entre el usuario y la aplicación web.

# 9. Bibliografía


[1] "¿Qué es y para qué sirve un API Manager? - Byteflair." [Online]. Available: https://byteflair.com/es/2016/07/que-es-para-que-sirve-api-manager/. [Accessed: 15-Feb-2019].

[2] "¿Qué es un API Manager? – Un poco de Java y +." [Online]. Available: https://unpocodejava.com/2013/03/27/que-es-un-api-manager/. [Accessed: 02-Feb-2019].

[3] "API Management: ¿qué es y para qué sirve? - Paradigma." [Online]. Available: https://www.paradigmadigital.com/dev/api-management-que-es-y-para-que-sirve/. [Accessed: 25-Feb-2019].

[4] "API Management Architecture - An introduction | Devoteam." [Online]. Available: https://nl.devoteam.com/en/blog-post/api-management-architecture-introduction/. [Accessed: 02-Mar-2019].

[5] "Top 10 Riesgos de Seguridad en la Web (OWASP) y como atenuarlos con API Management - Sensedia." [Online]. Available: https://sensedia.com/es/blog/apis/10-riesgos-seguridad-apis/. [Accessed: 10-Mar-2019].

[6] "OWASP Top 10 - 2017."

[7] "REST Security Cheat Sheet OWASP." [Online]. Available: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/REST_Security_Cheat_Sheet.md. [Accessed: 20-Mar-2019].

[8] "API Management Market by Solution, Services & Deployment Type | MarketsandMarkets." [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/api-management-market-178266736.html. [Accessed: 15-Mar-2019].

[9] "API Management Market Research Report- Global Forecast 2022|MRFR." [Online]. Available: https://www.marketresearchfuture.com/reports/api-management-market-2429. [Accessed: 15-Mar-2019].

[10] "API Management Market worth $5.1 billion by 2023." [Online]. Available: https://www.marketsandmarkets.com/PressReleases/api-management.asp. [Accessed: 17-Mar-2019].

[11] "Gartner Magic Quadrant." [Online]. Available: https://www.gartner.com/doc/reprints?id=1-4Y2SY8F&ct=180501&st=sb. [Accessed: 25-Mar-2019].

[12] "Plataforma de gestión de APIs Apigee | Apigee API Management Platform | Google Cloud." [Online]. Available: https://cloud.google.com/apigee-api-management/?hl=es. [Accessed: 26-Mar-2019].

[13] "API Gateway - CA Technologies." [Online]. Available: https://www.ca.com/us/products/apim/gateway.html. [Accessed: 19-Mar-2019].

[14] "API Management Platform SaaS - CA Technologies." [Online]. Available: https://www.ca.com/us/products/apim/saas.html. [Accessed: 21-Mar-2019].

[15] "CA Live API Creator: CA Technologies - España." [Online]. Available: https://www.ca.com/es/products/ca-live-api-creator.html. [Accessed: 21-Mar-2019].

[16] "API Developer Portal - CA Technologies." [Online]. Available:

https://www.ca.com/us/products/apim/developer-portal.html. [Accessed: 21-Mar-2019].
[17] “Mobile Gateway - CA Technologies.” [Online]. Available: https://www.ca.com/us/products/apim/mobile-api-gateway.html. [Accessed: 25-Mar-2019].
[18] “Microservice Gateway - CA Technologies.” [Online]. Available: https://www.ca.com/us/products/apim/microgateway.html. [Accessed: 25-Mar-2019].
[19] “Amazon API Gateway.” [Online]. Available: https://aws.amazon.com/es/api-gateway/. [Accessed: 04-Apr-2019].
[20] “Five Reasons to Consider Amazon API Gateway for Your Next Microservices Project - The New Stack.” [Online]. Available: https://thenewstack.io/five-reasons-to-consider-amazon-api-gateway-for-your-next-microservices-project/. [Accessed: 04-Apr-2019].
[21] “API Management: establecimiento de puertas de enlace de API | Microsoft Azure.” [Online]. Available: https://azure.microsoft.com/es-es/services/api-management/?ocid=AID754288&wt.mc_id=CFID0214. [Accessed: 05-Apr-2019].
[22] “Información general y conceptos clave de Azure API Management.” [Online]. Available: https://docs.microsoft.com/es-es/azure/api-management/api-management-key-concepts. [Accessed: 05-Apr-2019].
[23] “Why Azure API Management - biztalkbill.” [Online]. Available: https://www.biztalkbill.com/2018/07/23/azure-api-management/. [Accessed: 05-Apr-2019].
[24] “Azure API Management - Digital Marketplace.” [Online]. Available: https://www.digitalmarketplace.service.gov.uk/g-cloud/services/250014186005843. [Accessed: 05-Apr-2019].
[25] “Azure API Management.” [Online]. Available: https://www.reply.com/solidsoft-reply/en/content/azure-api-management. [Accessed: 05-Apr-2019].
[26] “Key Concepts - API Manager 2.6.0 - WSO2 Documentation.” [Online]. Available: https://docs.wso2.com/display/AM260/Key+Concepts. [Accessed: 07-Apr-2019].
[27] “API Management - Features.” [Online]. Available: https://wso2.com/api-management/features/. [Accessed: 07-Apr-2019].
[28] “Access the management API | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/system-administration/management-api-overview. [Accessed: 07-Apr-2019].
[29] “Edge built-in roles | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/system-administration/edge-built-roles. [Accessed: 06-Apr-2019].
[30] “Securing a proxy | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/security/api-security. [Accessed: 08-Apr-2019].
[31] “LDAP policy | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/reference/policies/ldap-policy. [Accessed: 08-Apr-2019].
[32] “Authenticating users and application clients | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-baas/security/authenticating-users-and-application-clients. [Accessed: 09-Apr-2019].

[33] “Last-mile security | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/security/last-mile-security. [Accessed: 07-Apr-2019].
[34] “TLS/SSL | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/system-administration/ssl. [Accessed: 10-Apr-2019].
[35] “Working with key value maps | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/cache/key-value-maps.html. [Accessed: 11-Apr-2019].
[36] “Use the API Monitoring Management API | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-monitoring/api. [Accessed: 15-Apr-2019].
[37] “Introduction to security reports | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/security/reports. [Accessed: 12-Apr-2019].
[38] “Content-based security | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/security/content-based-security.html. [Accessed: 16-Apr-2019].
[39] “Rate-limiting | Apigee Docs.” [Online]. Available: https://docs.apigee.com/api-platform/develop/rate-limiting. [Accessed: 17-Apr-2019].
[40] “Access Control Assertions - CA API Gateway - 9.2 - Documentación de CA Technologies.” [Online]. Available: https://docops.ca.com/ca-api-gateway/9-2/en/policy-assertions/assertion-palette/access-control-assertions. [Accessed: 19-Apr-2019].
[41] “Login: Authentication and Authorization Policies - CA Mobile API Gateway - 4.2 - Documentación de CA Technologies.” [Online]. Available: https://docops.ca.com/ca-mobile-api-gateway/4-2/en/mobile-policies/configure-policies/login-authentication-and-authorization-policies. [Accessed: 19-Apr-2019].
[42] “Transport Layer Security (TLS) Assertions - CA API Gateway - 9.2 - Documentación de CA Technologies.” [Online]. Available: https://docops.ca.com/ca-api-gateway/9-2/en/policy-assertions/assertion-palette/transport-layer-security-tls-assertions. [Accessed: 14-Apr-2019].
[43] “Remove Sensitive Data for Auditing - CA API Gateway - 9.3 - Documentación de CA Technologies.” [Online]. Available: https://docops.ca.com/ca-api-gateway/9-3/en/reference/pci-dss-implementation-guide/policy-construction-and-assertion-usage/remove-sensitive-data-for-auditing. [Accessed: 20-Apr-2019].
[44] “Manage Stored Passwords - CA API Gateway - 9.3 - Documentación de CA Technologies.” [Online]. Available: https://docops.ca.com/ca-api-gateway/9-3/en/security-configuration-in-policy-manager/tasks-menu-security-options/manage-stored-passwords. [Accessed: 21-Apr-2019].
[45] “About Message Auditing - CA API Gateway - 9.3 - Documentación de CA Technologies.” [Online]. Available: https://docops.ca.com/ca-api-gateway/9-3/en/policy-assertions/assertion-palette/logging-auditing-and-alerts-assertions/about-message-auditing. [Accessed: 22-Apr-2019].
[46] “Threat Protection Assertions - CA API Gateway - 9.3 - Documentación de CA Technologies.” [Online]. Available: https://docops.ca.com/ca-api-gateway/9-3/en/policy-assertions/assertion-palette/threat-protection-

assertions. [Accessed: 25-Apr-2019].
[47] “Service Availability Assertions - CA API Gateway - 9.3 - Documentación de CA Technologies.” [Online]. Available: https://docops.ca.com/ca-api-gateway/9-3/en/policy-assertions/assertion-palette/service-availability-assertions. [Accessed: 10-Apr-2019].
[48] “Control y administración del acceso a una API REST en API Gateway - Amazon API Gateway.” [Online]. Available: https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/apigateway-control-access-to-api.html. [Accessed: 24-Apr-2019].
[49] “Uso de certificados SSL del lado cliente para la autenticación por el backend - Amazon API Gateway.” [Online]. Available: https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/getting-started-client-side-ssl-authentication.html. [Accessed: 01-May-2019].
[50] “How to Enhance the Security of Sensitive Customer Data by Using Amazon CloudFront Field-Level Encryption | AWS Security Blog.” [Online]. Available: https://aws.amazon.com/es/blogs/security/how-to-enhance-the-security-of-sensitive-customer-data-by-using-amazon-cloudfront-field-level-encryption/. [Accessed: 01-May-2019].
[51] “Seguimiento, registro y monitoreo de una API de API Gateway - Amazon API Gateway.” [Online]. Available: https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/monitoring_overview.html. [Accessed: 03-May-2019].
[52] “Protection Capabilities - AWS WAF Security Automations.” [Online]. Available: https://docs.aws.amazon.com/es_es/solutions/latest/aws-waf-security-automations/capabilities.html. [Accessed: 04-May-2019].
[53] “Limitar las solicitudes de la API para mejorar el desempeño - Amazon API Gateway.” [Online]. Available: https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/api-gateway-request-throttling.html. [Accessed: 02-May-2019].
[54] “Directivas de autenticación de Azure API Management | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/es-es/azure/api-management/api-management-authentication-policies. [Accessed: 06-May-2019].
[55] “Configure Azure.” [Online]. Available: https://auth0.com/docs/integrations/azure-api-management/configure-azure. [Accessed: 06-May-2019].
[56] “Secure APIs using client certificate authentication in API Management - Azure API Management | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-mutual-certificates-for-clients. [Accessed: 06-May-2019].
[57] “Secure back-end services using client certificate authentication - Azure API Management | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-mutual-certificates. [Accessed: 06-May-2019].
[58] “Cómo usar valores con nombre en las directivas de Azure API Management | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/es-es/azure/api-management/api-management-howto-properties. [Accessed: 06-May-2019].
[59] “Introducción a firewall de aplicaciones web para Azure Application

Gateway | Microsoft Docs." [Online]. Available: https://docs.microsoft.com/es-es/azure/application-gateway/waf-overview. [Accessed: 08-May-2019].
[60] "Limitación avanzada de solicitudes con Azure API Management | Microsoft Docs." [Online]. Available: https://docs.microsoft.com/es-es/azure/api-management/api-management-sample-flexible-throttling. [Accessed: 08-May-2019].
[61] "WSO2Con EU 2015: Securing, Monitoring and Monetizing APIs." [Online]. Available: https://es.slideshare.net/wso2.org/3securing. [Accessed: 15-May-2019].
[62] "Extended Security with WSO2 API Management Platform." [Online]. Available: https://es.slideshare.net/wso2.org/extended-security-with-wso2-api-management-platform. [Accessed: 15-May-2019].
[63] "Working with Security - API Manager 2.6.0 - WSO2 Documentation." [Online]. Available: https://docs.wso2.com/display/AM260/Working+with+Security. [Accessed: 17-May-2019].
[64] "Configuring Transport Level Security - Administration Guide 4.4.x - WSO2 Documentation." [Online]. Available: https://docs.wso2.com/display/ADMIN44x/Configuring+Transport+Level+Security. [Accessed: 22-May-2019].
[65] "Masking Sensitive Information in Logs - Administration Guide 4.4.x - WSO2 Documentation." [Online]. Available: https://docs.wso2.com/display/ADMIN44x/Masking+Sensitive+Information+in+Logs. [Accessed: 23-May-2019].
[66] "Working with Encrypted Passwords - API Manager 2.6.0 - WSO2 Documentation." [Online]. Available: https://docs.wso2.com/display/AM260/Working+with+Encrypted+Passwords. [Accessed: 17-May-2019].
[67] "Viewing API Statistics - API Manager 2.6.0 - WSO2 Documentation." [Online]. Available: https://docs.wso2.com/display/AM260/Viewing+API+Statistics. [Accessed: 19-May-2019].
[68] "Alert Types - API Manager 2.6.0 - WSO2 Documentation." [Online]. Available: https://docs.wso2.com/display/AM260/Alert+Types. [Accessed: 21-May-2019].
[69] "Gateway Threat Protectors for API Manager - API Manager 2.6.0 - WSO2 Documentation." [Online]. Available: https://docs.wso2.com/display/AM260/Gateway+Threat+Protectors+for+API+Manager. [Accessed: 21-May-2019].
[70] "Working with Throttling - API Manager 2.6.0 - WSO2 Documentation." [Online]. Available: https://docs.wso2.com/display/AM260/Working+with+Throttling. [Accessed: 26-May-2019].
[71] "OWASP protections with Edge—defense in depth | Apigee Docs." [Online]. Available: https://docs.apigee.com/api-platform/faq/owasp-protection?tenant=apigee&hl=zh-tw&authuser=0. [Accessed: 26-May-2019].
[72] "SOLUTION BRIEF • CA API MANAGEMENT Enable and Protect Your Web Applications From OWASP Top Ten With CA API Management."

[73] “Use AWS WAF to Mitigate OWASP’s Top 10 Web Application Vulnerabilities,” 2017.
[74] “OWASP and Cloud Security Azure.” [Online]. Available: https://github.com/michaelsrichter/azure.owasp/blob/Public/OWASP_and_Cloud_Security1.1.pdf. [Accessed: 27-May-2019].