

Amenazas y vulnerabilidades en las Open API de TM Forum

Threats and Vulnerabilities in TM Forum Open APIs

Ing. Clayret Echenique Quintana^{1*},
Ing. Jorge Manuel Sotolongo Piedra ²

Recibido: 01/2026 | Aceptado: 03/2026 | Publicado: 07/2026

Resumen

El sector de las telecomunicaciones enfrenta desafíos de ciberseguridad distintivos derivados de su migración a arquitecturas nativas de la nube y su rol como eje central del entorno digital. En este contexto, las Open API de TM Forum, que materializan los principios de la arquitectura ODA, constituyen una superficie de ataque significativa al facilitar la interoperabilidad entre sistemas desacoplados. El presente estudio tuvo como objetivo identificar y analizar las principales amenazas y vulnerabilidades que afectan a estas interfaces, con el fin de contribuir a la concepción de futuros marcos de seguridad. Se empleó una metodología de revisión sistemática de literatura basada en el protocolo PRISMA, con búsquedas en seis bases de datos: IEEE Xplore, Scopus, Web of

^{1*} Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 ½, reparto Torrens, municipio Boyeros, La Habana, Cuba. cechenique@uci.cu

² Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 ½, reparto Torrens, municipio Boyeros, La Habana, Cuba. jorgemsp@uci.cu

Science, Google Scholar, ResearchGate y ScienceDirect y fuentes adicionales oficiales. De los registros identificados se incluyeron siete estudios tras aplicar criterios de inclusión y exclusión. Como principal aporte se construyó una taxonomía de correlación que integra las categorías STRIDE+P, las vulnerabilidades del OWASP API Top 10 2023 y los componentes de la arquitectura ODA, identificando 17 correlaciones significativas. Los resultados revelan que el 47 % presentan severidad crítica, concentradas en el API Gateway, la gestión de clientes y el control de acceso. Se concluye que la seguridad de estas APIs requiere controles de autorización granular y validación de esquemas a nivel de campo, constituyendo un fundamento empírico para marcos de seguridad específicos en ecosistemas ODA

Palabras clave: amenazas en las Open API, ODA, Open API, Open API de TM Forum, seguridad en las Open API

Abstract

The telecommunications sector faces distinctive cybersecurity challenges arising from its migration to cloud-native architectures and its role as a central axis of the digital environment. In this context, TM Forum's Open APIs, which materialize the principles of the ODA architecture, constitute a significant attack surface by enabling interoperability among decoupled systems. This study aimed to identify and analyze the main threats and vulnerabilities affecting these interfaces, in order to contribute to the design of future security frameworks. A systematic literature review methodology based on the PRISMA protocol was employed, with searches conducted across six databases: IEEE Xplore, Scopus, Web of Science, Google Scholar, ResearchGate, and ScienceDirect and additional official sources. From records identified, 7 studies were included after applying inclusion and exclusion criteria. As the main original contribution, a three-dimensional correlation taxonomy was constructed integrating STRIDE+P categories, OWASP API Top 10 2023 vulnerabilities, and ODA architecture components, identifying 17 significant correlations. The results reveal that 47% present Critical severity, concentrated in the API Gateway, customer management, and access control. It is concluded that

the security of these APIs requires granular authorization controls and field-level schema validation, providing an empirical foundation for specific security frameworks in ODA ecosystems.

Keywords: *Threats in Open API, ODA, Open API, TM Forum Open APIs, Open API Security*

Introducción

El sector de las telecomunicaciones enfrenta desafíos de ciberseguridad distintivos debido a su rol como eje central del entorno digital. Los proveedores de servicios son particularmente vulnerables a ciberamenazas derivadas de tensiones geopolíticas, incluyendo el espionaje y otras actividades maliciosas. También se enfrentan a riesgos tecnológicos propios de su migración a arquitecturas nativas de la nube, las cuales son menos conocidas y representan una amenaza para los servicios de red tanto nuevos como antiguos (Donegan, 2023). En este contexto, la estandarización y la adopción de arquitecturas abiertas surgen como pilares fundamentales para la innovación y la interoperabilidad. TM Forum ha sido pionero en el desarrollo de marcos de referencia y mejores prácticas para guiar esta evolución (TM Forum, 2025a).

Entre sus contribuciones más importantes se destaca el marco de referencia de Operaciones Digitales Abiertas (ODA, por sus siglas en inglés), concebido como un modelo arquitectónico basado en componentes desacoplados y reutilizables para construir ecosistemas de negocio ágiles. Para complementar esta visión, las Open API de TM Forum materializan los principios de la capa de implementación de ODA al proveer un conjunto estandarizado de especificaciones técnicas que permiten la integración fluida, segura y escalable entre sistemas, plataformas y socios en todo el ecosistema digital.

El creciente uso de las API (Interfaz de Programación de Aplicaciones, por sus siglas en inglés) en el sector de las telecomunicaciones ha facilitado la creación de nuevos productos, especialmente con el auge de los servicios en la nube y las aplicaciones móviles. Sin embargo, esta apertura introduce vulnerabilidades significativas que pueden ser explotadas por actores maliciosos.

Según el reporte *State of API Security* H2 en 2025, los principales problemas de seguridad de las API se deben a que el 80 % de los líderes de seguridad carecen de monitoreo continuo en tiempo real; el 33 % de las organizaciones sufrieron un incidente de seguridad debido a las API el último año, solo el 19 % están seguros de la precisión de su inventario de API y el 51 % de las organizaciones se encuentran todavía en etapas de planificación o en niveles básicos en su madurez de seguridad de API (Salt, 2025).

En este contexto, las Open API de TM Forum emergen como un estándar fundamental que impulsa la integración y la creación de un ecosistema de servicios interconectados en el sector de las telecomunicaciones. Estas interfaces de programación no solo facilitan la conectividad y la interoperabilidad entre servicios complejos, sino que también permiten una gestión integral de su ciclo de vida, incluso en entornos donde múltiples socios colaboran en la prestación del servicio.

Basadas en los principios del diseño REST y al ser tecnológicamente independientes, las Open API de TM Forum se han convertido en la base para una gestión de servicios sencilla y coherente. Pero esta estandarización, aunque esencial para la innovación, también extiende el alcance de las vulnerabilidades mencionadas; por ello, la seguridad de estas API se convierte en un pilar crítico para la fiabilidad y sostenibilidad del ecosistema digital.

Por este motivo, la presente investigación tiene como objetivo sistematizar las principales amenazas y vulnerabilidades a las que se exponen las Open API de TM Forum para contribuir a futuros trabajos de concepción de marcos de seguridad que integren las mejores prácticas y estándares de seguridad para las Open API de TM Forum.

Materiales y métodos

El presente estudio se fundamenta en una revisión sistemática de literatura orientada a identificar, analizar y sintetizar el estado actual de las amenazas y vulnerabilidades en las Open API de TM Forum. Para garantizar la rigurosidad metodológica del proceso de selección y análisis documental, se adoptó el protocolo PRISMA. Se realizó una

búsqueda en seis bases de datos científicas y repositorios académicos seleccionados por su cobertura en ingeniería informática, telecomunicaciones y ciberseguridad: IEEE Xplore, Scopus, Web of Science, Google Scholar, ResearchGate y ScienceDirect.

La estrategia de búsqueda se estructuró mediante la combinación de términos controlados y libres en inglés y español, utilizando operadores booleanos (AND, OR, NOT) y truncamientos. Se diseñaron tres bloques temáticos:

1. API y seguridad: «API security» OR «API vulnerability» OR «API threat» OR «Open API security» OR «API risk»
2. TM Forum / ODA: «TM Forum» OR «Open Digital Architecture» OR «ODA» OR «Open API TM Forum» OR «TMF Open API»
3. Amenazas y vulnerabilidades: «threat modeling» OR «STRIDE» OR «OWASP API Top 10» OR «cybersecurity» OR «vulnerability assessment» OR «risk assessment»

Se establecieron en la Tabla 1, los criterios de inclusión y exclusión para la selección de estudios:

Tabla 1. Criterios de inclusión y exclusión de la selección de estudios

Criterios de inclusión	Criterios de exclusión
• Documentos que aborden la seguridad de APIs en contextos de telecomunicaciones o arquitecturas abiertas. • Documentos que hagan referencia explícita a TM Forum, ODA o sus Open API. • Documentos que analicen amenazas, vulnerabilidades o riesgos de seguridad en APIs RESTful. • Documentos que propongan marcos, estándares o políticas de seguridad aplicables a APIs. • Publicaciones entre 2020 y 2025, para garantizar vigencia tecnológica. • Documentos de acceso completo o texto completo disponible.	• Documentos que no traten temas de seguridad en APIs. • Documentos centrados exclusivamente en APIs de otros sectores sin vinculación con telecomunicaciones. • Presentaciones comerciales, promocionales o documentos sin respaldo académico o técnico. • Duplicados entre bases de datos. • Documentos anteriores a 2020, salvo referencias normativas históricas esenciales. • Artículos en idiomas distintos al inglés o español, salvo que tuvieran traducción verificable.

La selección de documentos se realizó en tres fases, siguiendo el flujo PRISMA (Figura 1):

Fase 1 — Identificación: Se ejecutaron las búsquedas en las seis bases de datos, registrando el número de resultados por fuente y eliminando duplicados mediante el gestor bibliográfico Zotero y revisión manual cruzada.

Fase 2 — Revisión: Se evaluaron títulos y resúmenes contra los criterios inclusión/exclusión.

Fase 3 — Inclusión: Se recuperaron los textos completos de los documentos preseleccionados. Se aplicó nuevamente los criterios inclusión/exclusión al contenido completo, registrando las razones de exclusión. Los documentos finalmente incluidos fueron sometidos a extracción de datos.

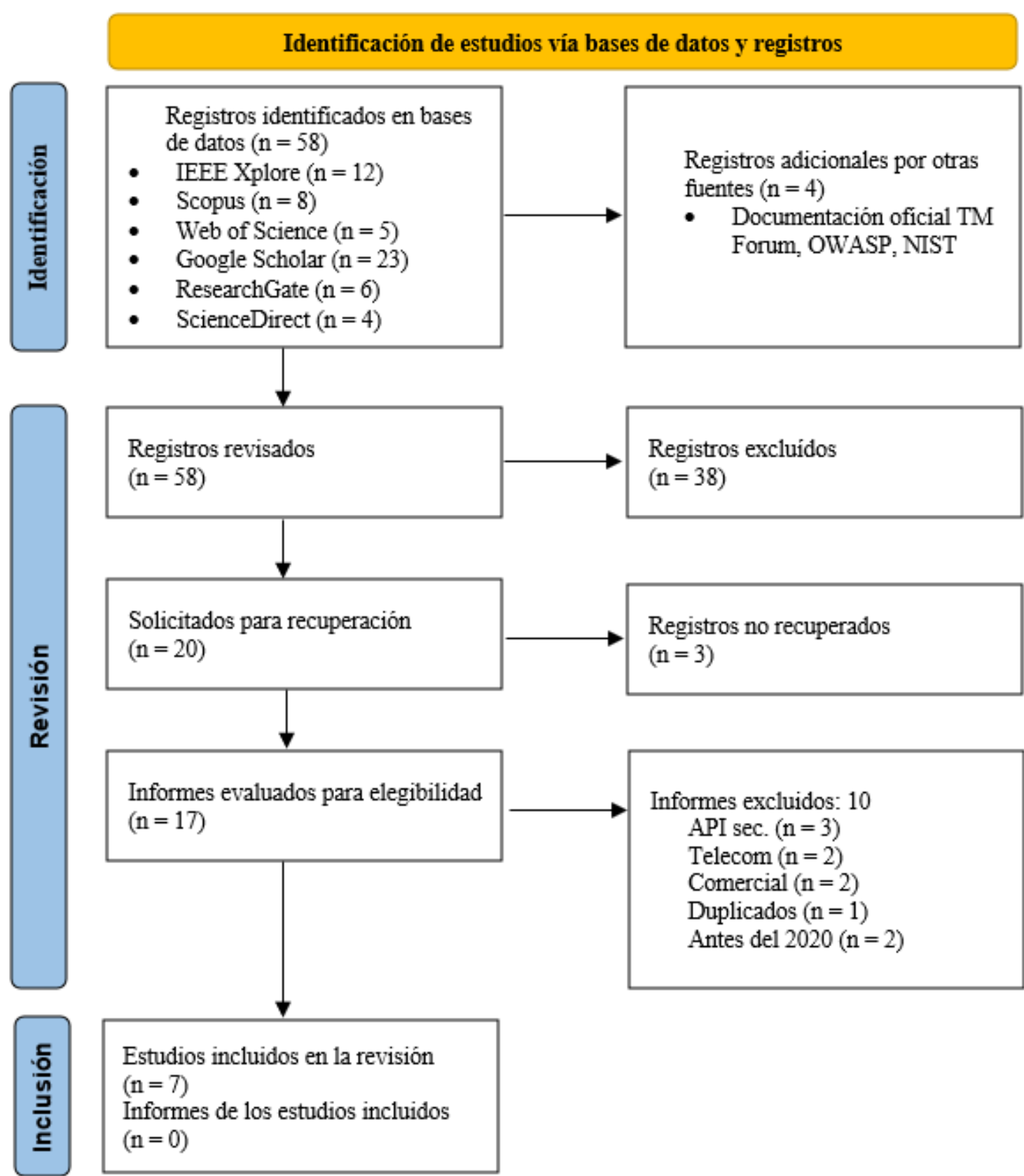


Figura 1. Flujo PRISMA para la selección de documentos

Luego, se aplicaron los siguientes métodos analíticos:

- **Análisis documental:** Revisión de la documentación oficial de TM Forum, estándares OWASP, NIST SP 800-53/63/228, y literatura académica seleccionada.
- **Análisis y síntesis:** Interpretación, procesamiento y categorización de la información extraída, estableciendo relaciones entre amenazas, vulnerabilidades de OWASP API Top 10 y controles de seguridad recomendados.

Resultados y discusión

Una API es un conjunto de reglas y definiciones que dictan cómo un *software* puede interactuar con otro. Actúa como un intermediario que facilita a diferentes sistemas de software comunicarse e intercambiar datos de manera fluida y estandarizada (Akintoye, 2025). Por ejemplo, desde una perspectiva más técnica y funcional, de forma especial en el contexto de la web, las API son el mecanismo que posibilita a varias aplicaciones compartir datos e interactuar mediante peticiones HTTP estructuradas (Akintoye, 2025). Durante la evolución de los servicios web, las API surgieron para satisfacer las demandas de conectividad fluida y rápido intercambio de datos (Jaiswal, 2025).

Dentro de las funciones esenciales de una API en la arquitectura de *software* moderna se encuentran (Akintoye, 2025; Garimilla, 2024):

- Promover la eficiencia: permiten a los desarrolladores integrar sistemas existentes en lugar de construir todo desde cero, lo que ahorra tiempo y recursos.
- Habilitar la conectividad: hacen posible que sistemas contruidos con tecnologías y plataformas completamente diferentes puedan trabajar juntos.
- Impulsar la innovación y la modularidad: ofrecen herramientas para construir soluciones creativas y compartimentar funcionalidades, lo que hace que el *software* sea más flexible, escalable y fácil de mantener.
- Acelerar el desarrollo: al facilitar la integración, reducen la complejidad y los ciclos de desarrollo.

Las API utilizan el Protocolo de Acceso a Objetos Simples (SOAP, por sus siglas en inglés), por su robustez, seguridad y estandarización, sobre todo en entornos empresariales complejos. Este es el protocolo más antiguo y rígido que se basa en mensajería XML. Se caracteriza por sus estrictos estándares de comunicación, contratos formalizados y robustas características de seguridad, siendo común en aplicaciones a nivel empresarial. Sin embargo, su complejidad a menudo ralentizaba el desarrollo (Akintoye, 2025; Jaiwal, 2025).

SOAP se considera, en gran medida, una tecnología heredada y el ecosistema digital ha experimentado una clara evolución desde SOAP

hacia arquitecturas REST. Esta transición fue impulsada por la necesidad de soluciones más ágiles, ligeras y escalables, en especial con el auge de las aplicaciones móviles y los servicios en la nube. Sus principios claves son el uso de métodos HTTP estándar (GET, POST, PUT, DELETE). La comunicación sin estado, donde cada petición contiene toda la información necesaria para ser procesada; el uso de formatos de datos ligeros, en particular JSON, y es altamente escalable y más sencillo de implementar (Akintoye, 2025; Jaiswal, 2025).

En la actualidad, GraphQL se menciona como una tendencia emergente y como una alternativa a REST, ya que su principal ventaja es la flexibilidad: permite a los clientes solicitar únicamente los datos que necesitan, evitando la sobrecarga de información (Jaiswal, 2025). Para obtener y manejar los datos, las APIs RESTful se organizan en torno a dos conceptos fundamentales (Akintoye, 2025):

1. **Recurso:** es el objeto central. Un recurso representa cualquier dato u objeto que puede ser accedido o manipulado como un usuario, un producto, una publicación o un libro. Cada recurso es identificado por una URI única que es la dirección para acceder a él.
2. **Endpoint:** es la URL específica donde un recurso puede ser accedido para realizar una operación. Un endpoint combina la URL base del servidor, la ruta del recurso y un método HTTP para definir una acción.

La interacción se produce cuando un cliente envía una petición a un endpoint y el servidor devuelve una representación del recurso, comúnmente en formato JSON.

Hoy en día, el rol de las API en aplicaciones web, móviles y en el Software como Servicio (SaaS, por sus siglas en inglés) es la columna vertebral que sostiene las aplicaciones modernas. La evolución hacia arquitecturas de microservicios y aplicaciones nativas de la nube fue impulsada por las API RESTful. Las API permiten desacoplar los componentes de un sistema, lo que facilita que diferentes equipos los desarrollen y desplieguen de forma independiente. En el entorno de la nube, las API RESTful son ideales porque consumen menos recursos computacionales y simplifican la escalabilidad horizontal en plataformas como Kubernetes. En cambio, en las aplicaciones móviles se

requiere un intercambio de datos rápido y eficiente para ofrecer una buena experiencia de usuario. Las API RESTful, con sus datos ligeros (JSON) y su menor latencia, son mucho más adecuadas para este propósito (Jaiswal, 2025).

Interface de Programación de Aplicaciones de TM Forum

Las Open API de TM Forum son el mecanismo que permite que todos los bloques y componentes de (ODA se conecten y funcionen como un sistema unificado y estable (Hou *et al.*, 2024; TM Forum, 2021a).

ODA es el marco de arquitectura de referencia de TM Forum, diseñado para reemplazar a los sistemas monolíticos con una colección de componentes de *software* desacoplados. Su objetivo principal es permitir a los proveedores de servicios de Internet operar como empresas digitales ágiles, similares a las grandes empresas de tecnología (Hou *et al.*, 2024; TM Forum, 2021a).

Las Open API de TM Forum tienen un estilo arquitectónico de API RESTful en formato JSON definidas con Especificaciones Open API (OAS, por sus siglas en inglés), y se utilizan para la orquestación y gestión de servicios y recursos entre Sistemas de Soporte al Negocio y Operaciones (BSS/OSS) (TM Forum, 2025b). Estas cumplen dos funciones esenciales dentro de la ODA (TM Forum, 2021a):

- Interfaces estandarizadas de componente: definen las interfaces de referencia mediante las cuales los componentes ODA (que residen en los bloques funcionales) exponen sus funcionalidades al exterior y se comunican entre sí. Este enfoque de exposición de API es lo que permite que la arquitectura sea platform-based y fomenta la evolución ágil.

Habilitador de interoperabilidad: al estar estandarizadas, las API permiten que componentes de diferentes proveedores o sistemas internos, propios o heredados, se conecten y trabajen juntos, lo que reduce de manera drástica la complejidad y el coste de la integración. El éxito de las Open API radica en que no son solo interfaces técnicas, sino que están ancladas a los marcos de referencia de TM Forum. Cada Open API se basa en un objeto de información de negocio definido en el Information Framework (SID) de TM Forum. Esta base semántica común asegura que, sin importar el proveedor,

el concepto de negocio intercambiado es consistente y universal. Por otra parte, mapean las funciones de los bloques ODA con los procesos de negocio definidos en el eTOM y esto asegura que las interfaces soportan flujos de trabajo de negocio de extremo a extremo (V. P. Mochalov *et al.*, 2019; TM Forum, 2021).

En la práctica, las Open API son fundamentales para la transformación digital, de manera especial en las áreas de orquestación, automatización y virtualización. En el contexto de una implementación de ODA, las APIs se utilizan para interconectar el nivel de gestión comercial con el nivel de producción de la red, por ejemplo, para automatizar la provisión de un servicio de conectividad.

Amenazas que exponen las Open API de TM Forum

En el documento IG1187 ODA Enterprise Risk Assessment v2.0 se identifican un conjunto de amenazas de seguridad relevantes mediante la adopción del modelo STRIDE+P, que abarcan las categorías de suplantación de identidad, manipulación de datos, repudio, divulgación de información, denegación de servicio, escalamiento de privilegios y riesgos de privacidad. Cada una de estas amenazas constituye un eje crítico en la gestión de riesgos de plataformas abiertas, pues comprometen la confidencialidad, integridad y disponibilidad de los servicios, además de la conformidad regulatoria en materia de protección de datos personales (TM Forum, 2021b).

En particular, la suplantación de identidad se relaciona con ataques donde un actor malicioso se hace pasar por un usuario o sistema legítimo; la manipulación de datos implica la alteración de información en tránsito o en reposo; el repudio refleja la ausencia de trazabilidad y registro que permita negar acciones realizadas; y la divulgación indebida de información corresponde a la exposición de datos sensibles de usuarios o empresas. Asimismo, la denegación de servicio busca interrumpir la disponibilidad de recursos mediante sobrecarga, el escalamiento de privilegios permite a un atacante obtener accesos no autorizados, y la categoría de privacidad enfatiza los riesgos de incumplimiento normativo, particularmente en relación con el Reglamento General de Protección de Datos (TM Forum, 2021b).

De manera más específica para el contexto de las Open API de TM Forum, el documento destaca que las interfaces de programación constituyen la principal superficie de ataque y recomienda adoptar como referencia las amenazas descritas en el OWASP API Top 10 (2019). Estas incluyen problemas de autorización a nivel de objeto, fallas en la autenticación, exposición excesiva de datos, ausencia de limitación de recursos o control de consumo, deficiencias en la autorización de funciones críticas, asignación masiva de atributos no controlados, configuraciones inseguras, vulnerabilidades de inyección, gestión inadecuada de activos y versiones y ausencia de registros y monitoreo efectivos (TM Forum, 2021b).

Dentro de este conjunto, las amenazas más críticas para las Open API de TM Forum son aquellas que impactan de manera directa en la autenticación, autorización y protección de datos. En particular, la exposición excesiva de datos constituye un riesgo central al poder revelar información personal o confidencial más allá de lo estrictamente necesario. La autenticación y autorización deficientes facilitan accesos indebidos a objetos o funciones sensibles, mientras que la ausencia de controles de consumo y gestión de versiones incrementa la vulnerabilidad frente a ataques de denegación de servicio. En este sentido, el marco de evaluación de riesgos de TM Forum evidencia que la seguridad de las Open API depende de la combinación de un análisis de amenazas estructurado bajo STRIDE+P y la adopción sistemática de las mejores prácticas propuestas por OWASP para API abiertas (Ariami *et al.*, 2024; Nikolov & Aleksieva-Petrova, 2025; TM Forum, 2021b).

Vulnerabilidades de las Open API de TM Forum

Las vulnerabilidades más comunes en las Open API de TM Forum coinciden con las descritas en el OWASP API Top 10 del 2019, según *IG1187 ODA Enterprise Risk Assessment* v2.0; aunque en esta guía referencian la lista del 2019, la autora adoptará las del 2023 que son la última versión estable: (Mazzei *et al.*, 2025; OWASP, 2023; Sharma, 2025):

1. Autorización rota a nivel de objeto (BOLA, por sus siglas en inglés): Las APIs a menudo exponen identificadores de objetos que pueden ser manipulados por un atacante.

2. Autenticación rota: Ocurre cuando los mecanismos de autenticación no se implementan de forma segura.
3. Autorización rota a nivel de propiedades del objeto: Cuando una API devuelve o permite modificar más campos de los necesarios o carece de controles por campo, puede exponer o alterar datos sensibles.
4. Consumo de recursos sin restricciones: Si la API no limita el número, tamaño o frecuencia de las solicitudes puede agotarse el rendimiento del sistema, generando denegaciones de servicio (DoS) o sobrecostos.
5. Autorización rota a nivel de función: Aparece cuando las funciones administrativas o de alto privilegio no están correctamente restringidas. Los atacantes pueden ejecutar acciones que deberían estar reservadas solo para ciertos roles.
6. Acceso sin restricciones a flujos de negocio sensibles: Incluso si no hay vulnerabilidades técnicas, los atacantes pueden abusar de la lógica de negocio expuesta por la API.
7. Falsificación de solicitudes del lado del servidor (SSRF): Ocurre cuando una API permite al usuario indicar direcciones URL externas o internas sin validación, haciendo que el servidor realice peticiones maliciosas.
8. Configuración de seguridad incorrecta: Incluye errores en cabeceras HTTP, permisos, políticas CORS, versiones inseguras, *endpoints* de depuración habilitados o claves expuestas.
9. Gestión inadecuada del inventario de APIs: Las organizaciones suelen perder el control de cuántas APIs tienen activas, sus versiones y a qué datos acceden.
10. Consumo inseguro de APIs: Cuando una aplicación consume APIs externas sin verificar su autenticidad, sin validar respuestas o sin controlar dependencias, puede convertirse en víctima de ataques.

Muchas APIs de TM Forum operan flujos de negocio complejos, interoperabilidad con terceros, versiones múltiples y escenarios de

consumo variado. Las vulnerabilidades como acceso sin restricciones a flujos de negocio sensibles y las de consumo inseguro son especialmente pertinentes. En este caso se carece de información que especifique que vulnerabilidades de las antes mencionadas afectan las Open API de TM Forum, por lo que se atribuirán en la investigación estas mismas vulnerabilidades expresadas en el Top 10 de OWASP (Friman *et al.*, 2023; V. Mochalov *et al.*, 2019; V. P. Mochalov *et al.*, 2020; Smidovych & Davydovskyi, 2022).

Seguridad y políticas de gestión de las Open API de TM Forum

La seguridad en la implementación de Open API debe abordarse bajo un enfoque integral que contemple tanto principios de diseño como mecanismos técnicos y de gestión. Se parte del supuesto de que el entorno es hostil, por lo que ninguna petición debe ser considerada confiable por defecto. Para ello, todo diseño, protocolo o API debe someterse a un proceso formal de modelado de amenazas y evaluación de riesgos, lo cual permite identificar la superficie de ataque y definir medidas de mitigación adecuadas (Ohtani *et al.*, 2021; Ranjan *et al.*, 2022; TM Forum, 2017).

Un aspecto clave es la delimitación de fronteras de confianza, con la minimización de la exposición de interfaces únicamente a las necesarias para el negocio, y aplicando un modelo de autorización granular tras una autenticación robusta. Dicha autenticación debe estar protegida mediante el uso de criptografía, ser obligatoria en todos los accesos, no poder deshabilitarse y quedar registrada en bitácoras inalterables. Asimismo, la autorización debe aplicarse después de autenticar, lo que asegura que cada usuario, sistema o plataforma solo acceda a los servicios y datos que le corresponden (Christian Chukwuemeka Ike *et al.*, 2021; Jangam *et al.*, 2022; Lafarga Poyo, 2025; TM Forum, 2017).

La protección de datos es transversal y debe garantizarse tanto en tránsito como en reposo y durante el procesamiento, evitando canales inseguros y empleando cifrado en todo momento. De igual forma, se recomienda segmentar los servicios en múltiples capas de defensa para contener posibles brechas, diseñar plataformas

libres de puntos únicos de fallo y aplicar configuraciones seguras por defecto (Babu, 2025; Gambo & Almulhem, 2025; Horbenko, 2025; Micheal, 2025; Gambo & Almulhem, 2025).

En entornos multiusuario o multiempresa es esencial asegurar el aislamiento entre inquilinos mediante mecanismos físicos, virtuales o lógicos, de forma que un compromiso no afecte a otros. El diseño también debe prever mantenibilidad, así como permitir actualizaciones y coexistencia de versiones de API sin comprometer la seguridad, además de garantizar la integridad y procedencia de los componentes a través inventarios protegidos y cadenas de confianza criptográficas (TM Forum, 2017).

De igual manera, la protección de la privacidad, especialmente de datos personales identificables, debe integrarse desde el inicio en cumplimiento con regulaciones como el Reglamento General de Protección de los Datos (GDPR), incorporando prácticas de minimización de datos, control de acceso, gestión de consentimiento y trazabilidad (Sun *et al.*, 2022; TM Forum, 2017).

En cuanto a la gestión de políticas de seguridad, cada plataforma define su propio dominio y frontera de confianza, siendo las API expuestas la superficie de ataque a proteger. Esto exige la implementación de mecanismos de control de acceso federado, autenticación multifactor —al menos de dos factores para datos sensibles—, asignación de políticas a roles y usuarios, así como la distribución y ejecución de dichas políticas en puntos de decisión y de cumplimiento dentro de la plataforma (Fernández-Martínez *et al.*, 2024; Lefebvre *et al.*, 2021; TM Forum, 2017).

Como referente tecnológico, el modelo de Perímetro Definido por Software (SDP) se plantea como un mecanismo ejemplar al permitir ocultar servicios, establecer conexiones cifradas de extremo a extremo y simplificar la gestión de autenticación, autorización y control de acceso. Este modelo se basa en tres componentes principales: el cliente SDP, encargado de verificar identidad y dispositivo; el controlador SDP, que actúa como intermediario de confianza configurando accesos en tiempo real; y el *gateway* SDP, que asegura la terminación

cifrada cercana a los servicios protegidos (Fernández-Martínez *et al.*, 2024; Lefebvre *et al.*, 2021; TM Forum, 2017).

De esta manera, las Open API no solo se protegen frente a ataques comunes como DDoS, *Man-in-the-middle*, OWASP Top 10 o amenazas persistentes avanzadas, sino que también se alinean con un marco sistemático de seguridad que incluye la definición de fronteras de confianza, identificación de amenazas, evaluación y cuantificación de riesgos residuales y establecimiento de políticas de acceso adaptadas a cada dominio (Christian Chukwuemeka Ike *et al.*, 2021; Min *et al.*, 2022; Sun *et al.*, 2022; TM Forum, 2017).

Taxonomía de amenazas y vulnerabilidades en Open API de TM Forum

A partir del análisis documental descrito en la metodología PRISMA, se identificó una carencia en la literatura especializada: la ausencia de un modelo de correlación que integre las categorías de amenazas del modelo STRIDE+P, las vulnerabilidades del OWASP API Security Top 10 y los componentes específicos de la arquitectura ODA. Para cubrir esta brecha, se diseñó una taxonomía propia de amenazas y vulnerabilidades mediante un proceso de correlación cruzada manual.

La construcción de la taxonomía siguió cuatro pasos metodológicos:

- Desagregación de fuentes. Se extrajeron las 7 categorías del modelo STRIDE+P del IG1187 ODA Enterprise Risk Assessment v2.0.0 (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege, Privacy Risks) y las 10 vulnerabilidades del OWASP API Security Top 10 2023.
- Correlación lógica. Se establecieron vínculos entre cada categoría STRIDE+P y las vulnerabilidades OWASP susceptibles de materializar dicha amenaza en un contexto de APIs RESTful. Se registraron únicamente las correlaciones con fundamento técnico explícito en la documentación analizada.
- Mapeo a componentes ODA. Cada correlación validada fue asignada al componente funcional de la arquitectura ODA donde la vulnerabilidad tendría mayor impacto.

• Jerarquización por severidad. Se aplicaron criterios propios de evaluación: impacto en continuidad del servicio telecom, exposición de datos personales o financieros, complejidad de detección y alineación con requisitos regulatorios. La severidad se clasificó en Crítica, Alta, Media y Baja.

La taxonomía resultante, presentada en la Tabla 2, identifica 17 correlaciones significativas entre las siete categorías STRIDE+P y 9 de las diez vulnerabilidades OWASP 2023. El identificador de las vulnerabilidades se corresponde con el valor numérico asignado en el apartado de vulnerabilidades de las APIs del presente artículo.

Tabla 2. Taxonomíade las amenazas y vulnerabilidades de las Open APIs de TM Forum

Categoría STRIDE+P	Vulnerabilidad OWASP 2023	Componente ODA Afectado	Severidad	Justificación
<i>Spoofing</i>	VUL_API2	API Gateway	Crítica	Compromiso de identidad en punto de entrada principal del ecosistema
	VUL_API8	Gestión de identidades	Alta	Configuración incorrecta de proveedores de identidad federada
<i>Tampering</i>	VUL_API1	Inventario de recursos	Crítica	Manipulación de identificadores de recursos de inventario de red
	VUL_API3	Catálogo del producto	Alta	Alteración de propiedades de productos y servicios catalogados
	VUL_API7	Servicios de orquestación	Alta	Modificación de flujos de orquestación internos mediante solicitudes forzadas
<i>Repudiation</i>	VUL_API8	Monitoreo y registro	Media	Ausencia de trazabilidad de acciones críticas por configuración deficiente
	VUL_API9	Inventario de APIs	Media	Falta de registro de APIs expuestas impide auditoría forense
<i>Information Disclosure</i>	VUL_API1	Gestión de clientes	Crítica	Exposición de datos de clientes por falla en autorización a nivel de objeto
	VUL_API3	Gestión de grupos	Crítica	Filtración de datos personales por sobreexposición de propiedades
	VUL_API6	Facturación e ingresos	Alta	Revelación de información financiera sensible por flujos desprotegidos
<i>Denial of Service</i>	VUL_API4	API Gateway	Crítica	Sobrecarga del gateway por consumo masivo de recursos sin control
	VUL_API7	Exposición de la red	Alta	Ataque a infraestructura interna desde API expuesta al exterior
<i>Elevation of Privilege</i>	VUL_API1	Control de acceso	Crítica	Escalamiento horizontal de privilegios entre objetos del mismo nivel
	VUL_API5	APIs administrativas	Crítica	Acceso a funciones administrativas sin autorización de rol
	VUL_API6	Ecosistemas de socios	Alta	Abuso de flujos de negocio para obtener accesos elevados en ecosistema multi-socios
<i>Privacy Risks</i>	VUL_API3	Perfil de clientes	Crítica	Exposición de PII violando GDPR y estándares PCI-DSS
	VUL_API9	Gestión de consentimiento	Alta	Pérdida de trazabilidad del consentimiento del usuario para tratamiento de datos

El análisis revela que el 47 % de las correlaciones presentan severidad Crítica, concentradas en tres ejes: divulgación de información, elevación de privilegios y denegación de servicios. La taxonomía permite identificar los componentes con mayor número de correlaciones

de severidad Crítica o Alta. En este caso, el API Gateway emerge como el componente más crítico. Este hallazgo justifica la recomendación de TM Forum de implementar mecanismos de SDP y Zero Trust en el perímetro de acceso.

A diferencia del modelo STRIDE original, la extensión STRIDE+P incorpora explícitamente los riesgos de privacidad. La taxonomía demuestra que la categoría: Privacidad no opera de forma aislada, sino que se manifiesta principalmente a través de la vulnerabilidad API3 en los componentes: Perfil de clientes y Gestión de consentimiento. Esto sugiere que la protección de datos personales en las Open API de TM Forum no puede abordarse únicamente con controles de acceso, sino que requiere validación de esquemas de datos a nivel de campo y mecanismos de minimización de datos desde el diseño.

Se reconocen dos limitaciones metodológicas: la correlación se basa en análisis documental, sin validación empírica mediante pruebas de penetración en implementaciones reales de Open API TM Forum; y la severidad asignada es relativa al contexto de telecomunicaciones y puede variar en implementaciones específicas. Estas limitaciones definen el trabajo futuro propuesto en las conclusiones.

Conclusiones

El estudio de la seguridad de las Open API de TM Forum, evidencian la creciente necesidad de abordar las vulnerabilidades inherentes a este tipo de arquitecturas. El estudio reveló la existencia de amenazas significativas que exponen estas interfaces de programación. Particularmente, en el análisis de las vulnerabilidades se destacó la criticidad de los riesgos asociados a estas interfaces. La naturaleza abierta de estas APIs, diseñada para fomentar la interoperabilidad y la innovación en el sector de las telecomunicaciones, introduce simultáneamente puntos de exposición que pueden ser explotados por agentes maliciosos.

Es por ello que, dentro de las recomendaciones de seguridad de TM Forum y por otra parte las de OWASP principalmente indican que toda API debe someterse a un proceso formal de modelado de amenazas y evaluación de riesgos, lo cual permite identificar la superficie de ataque y definir medidas de mitigación adecuadas. El método

que se plantea para modelar amenazas en los marcos de seguridad es STRIDE+P según IG1186, IG1187 e IG1195, y la adopción sistemática de las mejores prácticas propuestas por OWASP para Open APIs.

Parte de esas buenas prácticas, se consideran en los proyectos y normativas como *OWASP API Security Project*, *OWASP Application Security Verification Standard (ASVS)*, y las series *NIST SP 800-228*, *800-63* y *800-53 Rev 5*. Estos recursos proporcionan un marco integral para diseñar e implementar controles de seguridad que aborden las vulnerabilidades más comunes y críticas en entornos como en los que convergen las Open API de TM Forum. Es por ello que se deben tener en cuenta a la hora de seleccionar los controles de seguridad para este tipo de API.

La taxonomía presentada constituye un fundamento empírico para el desarrollo de futuros marcos de seguridad específicos para las Open API de TM Forum. Este modelo de correlación permite localizar amenazas arquitectónicas concretas, priorizar controles por componente y dimensionar el impacto regulatorio en función del tipo de datos expuestos.

Referencias bibliográficas

Akintoye, A. (2025). *API Development with Laravel: A Quick Start Guide*. Apress. <https://doi.org/10.1007/979-8-8688-1576-8>

Ariami, D., Windarta, S., Priambodo, D. F., & Amiruddin, A. (2024). Combining STRIDE and Penetration Testing for Improved Web Application Security: A Case Study on Stored XSS Vulnerabilities. 2024 *Beyond Technology Summit on Informatics International Conference (BTS-I2C)*, 332-337. <https://doi.org/10.1109/BTS-I2C63534.2024.10942270>

Babu, M. (2025). *Enhancing Cloud Security: Implementing and Evaluating the Zero Trust Architecture with Firebase Services and Advanced Encryption Algorithms* [Masters, Dublin, National College of Ireland]. <https://norma.ncirl.ie/7426/>

Christian Chukwuemeka Ike, Adebimpe Bolatito Ige, Sunday Adeola Oladosu, Peter Adeyemo Adepoju, Olukunle Oladipupo Amoo, & Adeoye Idowu Afolabi. (2021). Redefining zero trust architecture in cloud networks: A conceptual shift towards granular, dynamic access control and policy enforcement. *Magna Scientia Advanced Research and Reviews*, 2(1), 074-086. <https://doi.org/10.30574/msarr.2021.2.1.0032>

- Csoma, M.-I., Koné, B., Botez, R., Ivanciu, I.-A., Kora, A., & Dobrota, V. (2020). *Management and Orchestration for Network Function Virtualization: An Open Source MANO Approach*. 2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet), 1-6. <https://doi.org/10.1109/RoEduNet51892.2020.9324847>
- Donegan, P. (2023). *Cybersecurity strategies: Risk management moves firmly into the telco spotlight*. <https://inform.tmforum.org/research-and-analysis/reports/cybersecurity-strategies-risk-management-moves-firmly-into-the-telco-spotlight>
- Fernández-Martínez, C., Bikos, A., Verikoukis, C., & Siddiqui, S. (2024). *Trusted Access to 6G Testbeds Through a Security Intent-Driven Software-Defined Perimeter Framework*. 2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 1-6. <https://doi.org/10.1109/CAMAD62243.2024.10942785>
- Gambo, M. L., & Almulhem, A. (2025). Zero Trust Architecture: A Systematic Literature Review. *Journal of Network and Systems Management*, 34(1), 25. <https://doi.org/10.1007/s10922-025-09998-x>
- Garimilla, M. (2024). THE ART OF API DESIGN: BEST PRACTICES FOR MODERN SOFTWARE DEVELOPMENT. *International Journal of Engineering and Technology Research (IJETR)*, 9(2), 229-239. <https://doi.org/10.5281/ZENODO.13772395>
- Horbenko, Y. (2025). Secure Front-End Automation Framework: A Novel Approach to Client-Side Data Encryption and Zero Trust API Interaction. *Asian Journal of Research in Computer Science*, 18(6), 177-193. <https://doi.org/10.9734/ajrcos/2025/v18i6690>
- Hou, X., Zhao, Y., & Wang, H. (2024). *Voices from the Frontier: A Comprehensive Analysis of the OpenAI Developer Forum*. arXiv. <https://doi.org/10.48550/arXiv.2408.01687>
- Jaiswal, I. A. (2025). *THE EVOLUTION OF WEB SERVICES AND APIS: FROM SOAP TO RESTFUL DESIGN*. 14.
- Jangam, S. K., Karri, N., & Muntala, P. S. R. P. (2022). Advanced API Security Techniques and Service Management. *International Journal of Emerging Research in Engineering and Technology*, 3(4), 63-74. <https://doi.org/10.63282/3050-922X.IJERET-V3I4P108>
- Lafarga Poyo, F. (2025). *Zero trust framework for microservices security: Analysis, design, and implementation*. <https://hdl.handle.net/20.500.14468/29428>

- Lefebvre, M., Nair, S., Engels, D. W., & Horne, D. (2021). *Building a Software Defined Perimeter (SDP) for Network Introspection*. 2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 91-95. <https://doi.org/10.1109/NFV-SDN53031.2021.9665152>
- Mazzei, C., Firoamant, D., Quagliarini, A., Salatino, C., Reale, A., & Brunetti, F. (2025). *The Open Chain Framework for the “Management by Projects” in Fiber-only Network Wholesale Telcos*.
- Micheal, D. (2025). *Digital Identity and Privacy in API-Driven Environments: Combining Blockchain, Cryptography, and Forensic Techniques*.
- Min, N. M., Visoottiviseth, V., Teerakanok, S., & Yamai, N. (2022). *OWASP IoT Top 10 based Attack Dataset for Machine Learning*. 2022 24th International Conference on Advanced Communication Technology (ICACT), 317-322. <https://doi.org/10.23919/ICACT53585.2022.9728969>
- Mochalov, V. P., Bratchenko, N. Yu., & Yakovlev, S. V. (2019). *Process-Oriented Management System for Infocommunication Networks and Services Based on TM Forum Framework*. 2019 International Russian Automation Conference (RuAutoCon), 1-6. <https://doi.org/10.1109/RUSAUTOCON.2019.8867619>
- Nikolov, L., & Aleksieva-Petrova, A. (2025). Threat modeling based on STRIDE framework for web application. *AIP Conference Proceedings*, 3308(1), 030007. <https://doi.org/10.1063/5.0272341>
- Ohtani, M., Omata, S., & Kanishima, K. (2021). *Development of Reference Model for Enterprise Architecture Leveraging TM Forum Assets*. 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), 210-213. <https://doi.org/10.23919/APNOMS52696.2021.9562626>
- OWASP. (2023). *OWASP API Security Project | OWASP Foundation*. <https://owasp.org/www-project-api-security/>
- Ranjan, P., Khunger, A., Veera, C., Satya, V., & Dahiya, S. (2022). *Threat Modeling and Risk Assessment of APIs in Fintech Applications*. 2, 44-61. <https://doi.org/10.56472/25832646/JETA-V2I2P108>
- Salt. (2025). *Salt Security: State of API Security Report*. <https://content.salt.security/Gartner-Voice-of-the-Customer.html>
- Sharma, A. (2025). *5G APIs security and privacy threat analysis within 5G technology domain*. <https://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-25387>
- Sun, R., Wang, Q., & Guo, L. (2022). *Research Towards Key Issues of API Security*. En W. Lu, Y. Zhang, W. Wen, H. Yan, & C. Li (Eds.), *Cyber Security* (pp. 179-192). Springer Nature. https://doi.org/10.1007/978-981-16-9229-1_11

TM Forum. (2017). *Platform Security and Policy Management R17.0.1 (TR263D)* – TM Forum. <https://www.tmforum.org/resources/technical-report/tr263d-platform-security-and-policy-management-r17-0-1/>

TM Forum. (2021a). *IG1167 ODA Functional Architecture Exploratory Report v5.1.0*. TM Forum. <https://www.tmforum.org/resources/exploratory-report/ig1167-oda-functional-architecture-exploratory-report-v5-1-0/>

TM Forum. (2021b). *IG1187 ODA Enterprise Risk Assessment v2.0.0*. TM Forum. <https://www.tmforum.org/resources/standard/ig1187-oda-enterprise-risk-assessment-v2-0-0/>

TM Forum. (2025a). *About TM Forum* | TM Forum. <https://www.tmforum.org/about/>

TM Forum. (2025b). *OpenAPI Directory*. <https://www.tmforum.org/oda/open-apis/directory>

