IoT security

Ana Poveda Díaz
Sofía Roncal Loyola

Índice

¿Qué es IoT?

El Internet of Things (IoT) es una red de dispositivos físicos conectados a internet que pueden recopilar, enviar y actuar sobre datos sin intervención humana directa. Estos dispositivos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas. Ejemplos: sensores, electrodomésticos, relojes inteligentes, coches...

Ejemplo cotidiano: una nevera que avisa cuando falta leche o un reloj que monitoriza tus pulsaciones.

Fuente:https://www.oracle.com/es/internet-of-things/

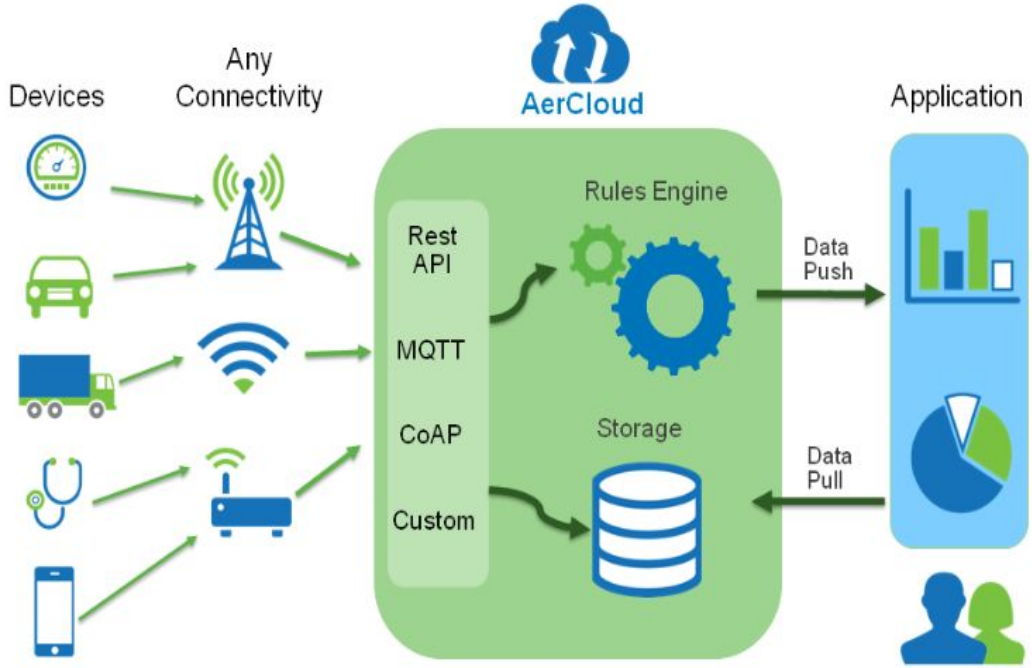
Aplicaciones de IoT

Algunas aplicaciones de Iot son:

- **Ámbito doméstico:** asistentes virtuales (Amazon Alex, Google Home), termostatos inteligentes.
- **Sector industrial:** mantenimiento predictivo de maquinaria, automatización.
- **Retail:** seguimiento de inventario en tiempo real.
- **Transporte y logística:** drones o robots de reparto, sensores RFID para rastreo de mercancías en tiempo real.
- **Sector agrícola:** sensores de humedad, control de riego inteligente.
- **Ciudades y edificios inteligentes:** semáforos adaptativos, gestión de residuos.
- **Ámbito sanitario:** Apple Watch / Fitbit para monitorear el ritmo cardiaco, monitores de glucosa.
- **Medio ambiente:** sensores de calidad del aire, detección de incendios.

https://builtin.com/articles/iot-devices

Arquitectura de un Sistema IoT



- **Dispositivos:** dispositivo con el que vamos a medir o interactuar con él.
- **Conectividad:** Medio de comunicación, cómo vamos a comunicar el HW, ya sea por red o de forma inalámbrica. (LoRaWan, zigBee, WiFi, bluetooth, etc.)
- **Protocolos de comunicación:** lenguaje para comunicar el HW y el SW. HTTP, fiware, MQTT, API REST, CoAp, etc.)
- **Plataformas Software**: para tratar los datos recogidos por nuestros sensores y almacenarlos.
- **Servicios**: son los servicios que ofrecen las plataformas como mostrar los datos recogidos, mandar avisos cuando se detecte un evento o la interconexión con otras plataformas o simplemente.

https://es.digi.com/blog/post/the-4-stages-of-iot-architecture

Principales vulnerabilidades en Iot

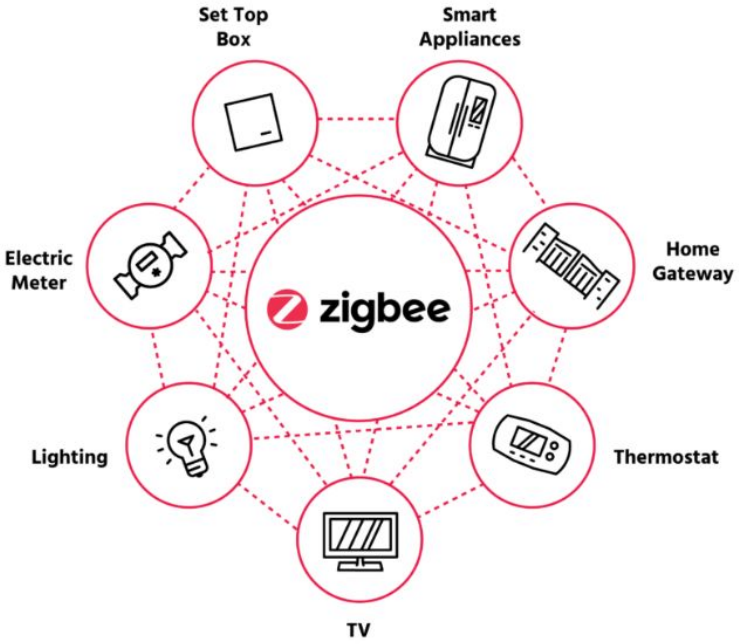
- Mecanismos de autenticación y autorización insuficientes
- Contraseñas débiles: Claves por defecto de fábrica
- Protocolos de comunicación inseguros: uso de comunicación de texto simple como TCP.
- Falta de capacitación de los usuarios: es posible que los proveedores de IoT no brinden la concientización sobre seguridad adecuada a sus usuarios.
- Ataques DoS: Consisten en saturar la red IoT o explotar fallos de software usando múltiples dispositivos, impidiendo el funcionamiento normal del sistema.

Protocolos de Conectividad en Iot

ZigBee

Protocolo de comunicación inalámbrica diseñado especialmente para aplicaciones de IoT que requieren **bajo consumo de energía**, **corto alcance** y transmisión de datos sencilla.

- Zigbee funciona según la especificación **IEEE 802.15.4**.
- Usa **redes tipo malla**: los dispositivos se comunican entre ellos y extienden el alcance.
- Ideal para redes domésticas o cerradas.



Smart Home

ZigBee

Seguridad:

- **Cifrado AES-128**: Protege los mensajes para que nadie pueda leerlos si los intercepta.

La versión más reciente, **Zigbee PRO 2023**, ha reforzado varios aspectos clave:

- **Autenticación**: asegura que sólo los dispositivos autorizados puedan comunicarse con la red.
- **Certificados**: valida si un dispositivo es legítimo o si alguien intenta introducir uno falso.
- **Trust Center Swap-Out**: permite reemplazar el coordinador de la red sin comprometer la seguridad de la red ni tener que reiniciar los dispositivos conectados.

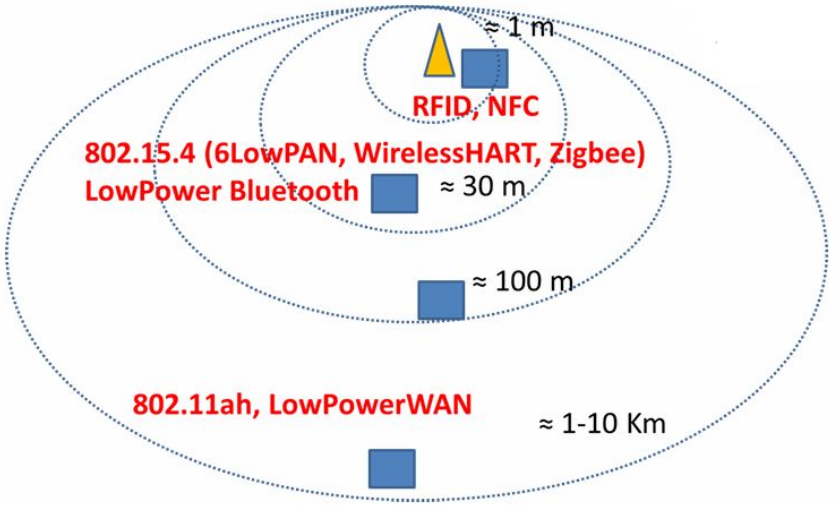
ZigBee: Vulnerabilidad vs Solución

Capa	Vulnerabilidad	Solución
Capa Física	**Interferencia o jamming en la banda 2.4 GHz** Emite señales de radio para interrumpir o bloquear las comunicaciones inalámbricas entre dispositivos.	Uso de canales alternos y cambios automáticos de canal para mitigar interferencias.
Capa Aplicación	**Accesos no autorizados por configuración débil:** Expone a la red a un control indebido por parte de dispositivos no autorizados	Uso de Access Control Lists (ACLs) y separación de roles: qué nodo puede enviar o recibir qué comandos.
General	Falta de cifrado en datos transmitidos en algunas implementaciones antiguas	Uso de cifrado AES-128 por defecto. En Zigbee PRO 2023, se refuerza la generación y gestión de claves seguras con DTLS en capas superiores si es requerido.

LoraWan

LoRaWAN (Long Range Wide Area Network) es un protocolo de red de baja potencia y largo alcance diseñado para aplicaciones de IoT. Se basa en la tecnología LoRa (Long Range).

- Permite la comunicación inalámbrica de dispositivos a través de largas distancias utilizando muy poca energía.
- No es adecuado para enviar una trama cada pocos minutos. No se recomienda para la transmisión en tiempo real.
- Topología estrella: los dispositivos (nodos) se comunican directamente con una o varias gateways que a su vez se conectan a un servidor central (network server).
- Opera en bandas sin licencia.



LoraWan: Vulnerabilidad vs Solución

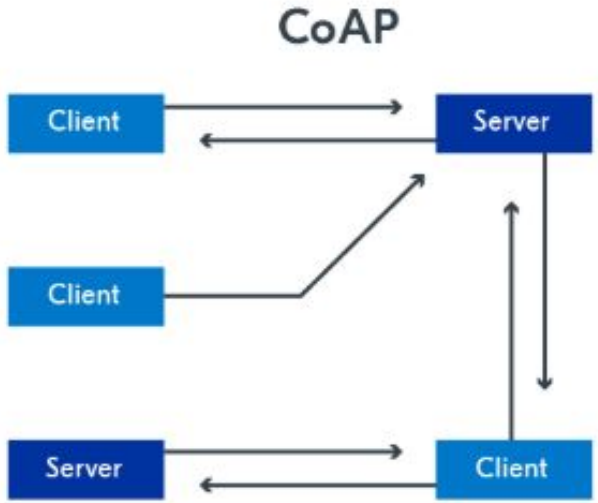
Capa	Vulnerabilidad	Solución
Capa aplicación	**Ataques de repetición (replay attacks)** Mal manejo del contador de tramas y verificación de mensajes.	Validación estricta de contadores de frames.
Capa aplicación	**Robo de claves (key extraction)** Las claves están mal protegidas en el dispositivo físico.	Uso de claves únicas por dispositivo.
Capa física	**Ataques de denegación de servicio (DoS)** Jamming RF o saturación de canales.	Usa múltiples canales y saltos de frecuencia automáticos.

Protocolos de Comunicación en Iot

CoAP: Constrained Application Protocol (RFC 7252)

CoAP es un protocolo diseñado específicamente para dispositivos IoT con recursos limitados.

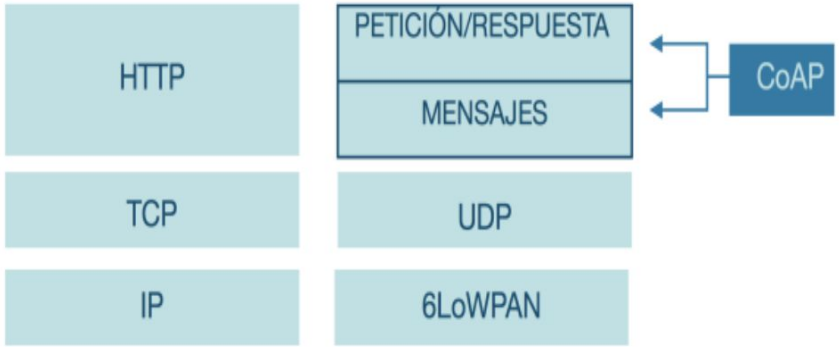
- Protocolo de transporte UDP.
- Modelo Cliente-Servidor, similar a HTTP.
- Comunicación asíncrona y directa entre nodos.
- Diseñado especialmente para trasladar el modelo de requests y responses de HTTP a dispositivos y redes con recursos limitados (poca memoria, almacenamiento, ancho de banda).



https://www.gotoiot.com/pages/articles/coap_intro/index.html

CoAP: Constrained Application Protocol (RFC 7252)

- CoAP se divide en dos subcapas principales:
  - Capa Message: maneja la comunicación con UDP y el envío/recepción de mensajes asincrónicos.
  - Capa Request/Response: gestiona las peticiones y sus respuestas, similar al modelo HTTP.
- Similitudes con HTTP:
  - Métodos: GET, POST, PUT, DELETE
  - Respuestas:
    - Success: 2.XX (2.03 valid, 2.05 Content, etc)
    - Client error: 4.XX (4.03 Forbide, 4.04 Not Found, etc)
    - Server error: 5.XX (5.01 Not Implemented, etc)

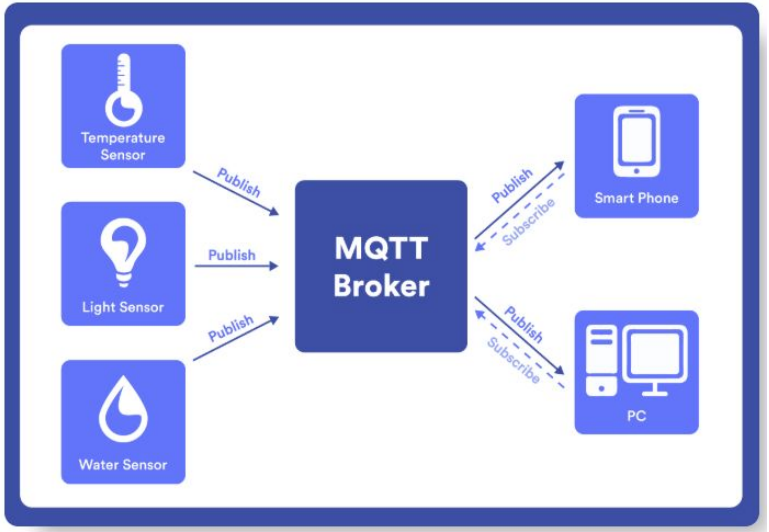


https://www.gotoiot.com/pages/articles/coap_intro/index.html

CoAP: Vulnerabilidad vs Solución

Capa	Vulnerabilidad	Solución
Capa de transporte	**No hay cifrado ni autenticación por defecto.** CoAP no cifra por defecto, dejando datos expuestos a interceptación.	Usar **DTLS** (Datagram Transport Layer Security) para cifrado de extremo a extremo.
Capa de Red	**Suplantación de identidad (spoofing)** Un atacante puede enviar mensajes falsos suplantando un cliente o servidor.	Autenticación mutua con DTLS.
Capa de aplicación	**Reenvío de mensajes (Replay Attacks)** Los mensajes pueden ser reenviados por un atacante, causando comportamientos no deseados.	Usar mecanismos **antireplay** de DTLS y t**okens únicos por mensaje**.
Capa de aplicación	**Recursos expuestos a escaneo** Recursos CoAP pueden ser fácilmente descubiertos y explorados por atacantes.	Configurar respuestas mínimas a /.well-known/core y aplicar autenticación.

MQTT: Message Queuing Telemetry Transport



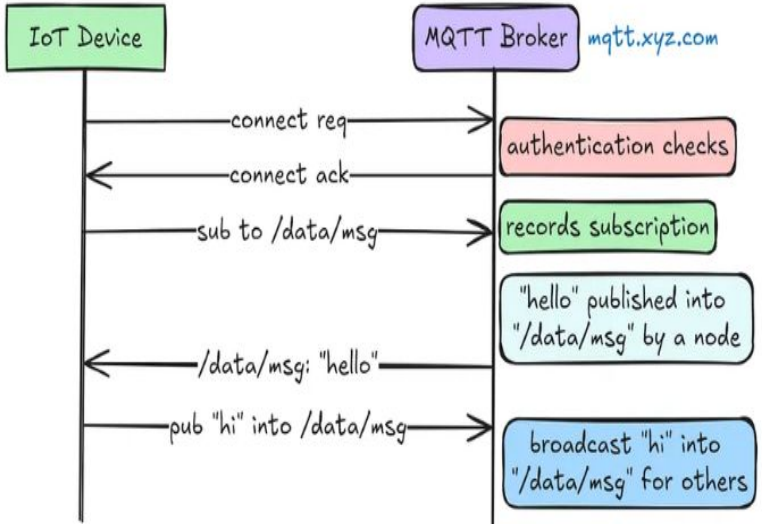
- Dómotica
- Protocolo de transporte: TCP
- Modelo de publicación/suscripción.
- Comunicación asíncrona, mediada por un broker
- Fiabilidad alta
- Tiene baja complejidad, ocupa poco espacio en el código y consume poco ancho de banda de red para la mensajería.
- Los publicadores envían mensajes sin conocer a los suscriptores, quienes los reciben sólo si están suscritos a temas específicos. Este sistema permite una comunicación escalable de uno a muchos, con independencia entre emisores y receptores.

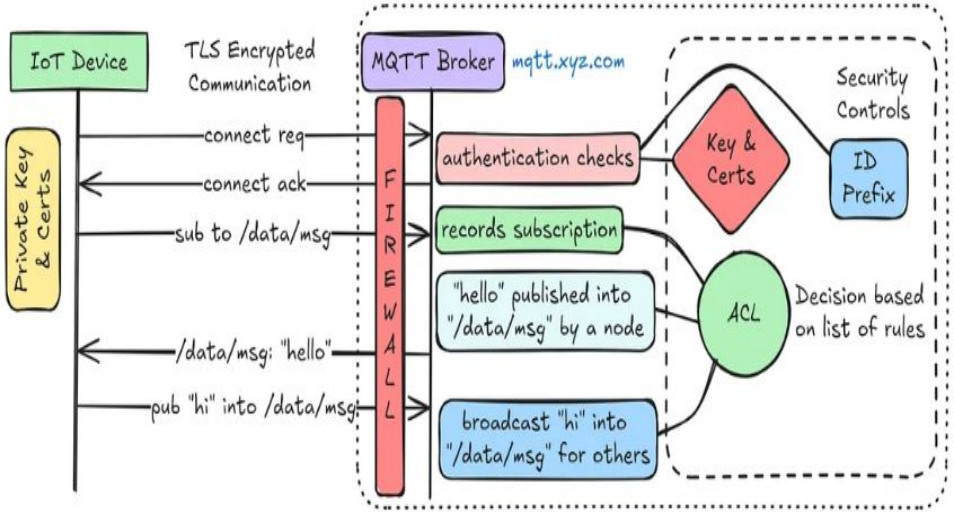
https://devopedia.org/mqtt

MQTT: Vulnerabilidad vs Solución

Capa	Vulnerabilidad	Solución
Capa de red	**Acceso no autorizado a la red IoT** MQTT funciona sobre IP, por lo que intrusos podrían acceder si la red no está protegida.	- Usar **firewalls** para bloquear conexiones y puertos innecesarios. - **Cerrar puertos UDP** para reducir riesgos. - Implementar **VPNs** e **IPsec (Internet Protocol Security)** para cifrar el tráfico y limitar el acceso.
Capa de transporte	**Transmisión de datos en texto(por TCP/WebSocket)** Sin cifrado, datos sensibles, como credenciales, pueden ser interceptados por un atacante.	Implementar **TLS (Transport Layer Security)** para cifrado extremo a extremo, Esto impide que un atacante vea o altere los mensajes MQTT durante la transmisión.
Capa de aplicación	**Autenticación débil o insuficiente** Si se utilizan credenciales predeterminadas o débiles, un atacante podría conectarse como dispositivo legítimo.	Usar **autenticación por usuario y contraseña segura**, **tokens**, y en MQTT 5.0, habilitar la **autenticación mejorada (Enhanced Authentication)**.
Capa de aplicación	**Acceso no controlado a temas** Los dispositivos pueden publicar o leer de temas sin autorización si no se establece un control adecuado.	Establecer **políticas de control de acceso** para definir quién puede publicar o suscribirse a determinados temas. - **Access Control Lists (ACLs)** para definir quién puede publicar o suscribirse a qué temas. - **Utilizar Node IDs** con un prefijo único para asegurar que solo dispositivos autorizados tengan acceso.

MQTT: Vulnerabilidades





Discusión:

Escenario 1: Edificio Inteligente

Un sistema de luces y sensores de presencia debe operar en una red cerrada, con bajo consumo y respuestas rápidas.

Pregunta: ¿Qué protocolo elegirías: LoraWan ,Zigbee, MQTT o CoAP?

Discusión:

Escenario 2:

Si un sistema IoT necesita transmitir datos de forma confiable y segura, pero con bajo consumo energético,

Pregunta: ¿preferirías MQTT o CoAP? ¿Por qué?

Pregunta para reflexionar:

¿Creen que hay un único protocolo ideal para todo tipo de dispositivos IoT? ¿O siempre dependerá del contexto? ¿Qué factores clave deberían considerarse?

Gracias