Análisis Firmware

CIBERSEGURIDAD EN ENTORNOS DE LAS TECNOLOGÍAS DE LA INFORMACIÓN

MODULO ANALISIS FORENSE

ALUMNO: XINWEI WU

Resumen Ejecutivo

Este informe presenta los resultados del análisis forense digital realizado sobre el firmware de un dispositivo, con el objetivo de identificar los sistemas de archivos, servicios y usuarios presentes en el sistema, así como evaluar posibles vulnerabilidades o configuraciones que puedan representar riesgos de seguridad.

Hallazgo Principales de la bombilla:

1. **Bombilla IoT:**
    - Información obtenida: No se pudo extraer información relevante del firmware debido a las restricciones impuestas por el desarrollador.

Hallazgos Principales de la Cámara

1. **Sistemas de Archivos:**
    - Se identificaron dos sistemas de archivos principales: Squashfs (comprimido y de solo lectura) y JFFS2 (para almacenamiento flash con capacidad de lectura y escritura). Esta combinación es común en dispositivos embebidos, ya que separa el sistema operativo de los datos modificables.

2. **Servicios en Ejecución:**
    - Se detectaron varios servicios críticos, incluyendo mdev (gestión de dispositivos), miio_client (comunicación con dispositivos IoT) y miio_avstreamer (transmisión de audio y video). Estos servicios sugieren que el dispositivo está diseñado para funcionar en un entorno IoT con capacidades multimedia.

3. **Usuarios del Sistema:**
    - Se identificaron 11 usuarios en el archivo /etc/passwd, siendo el usuario root el más crítico debido a sus privilegios totales. La mayoría de los usuarios tienen shells deshabilitadas, lo que reduce el riesgo de accesos no autorizados.

4. **Tipo de Análisis:**
    - El análisis realizado es un análisis forense digital, que utiliza herramientas como binwalk para extraer y examinar el contenido del firmware. Este tipo de análisis es esencial para identificar vulnerabilidades, preservar evidencias y comprender el funcionamiento del sistema.

1. Introducción

El presente informe tiene como objetivo presentar los resultados del análisis forense realizado sobre el firmware de una cámara y una bombilla. Este análisis se llevó a cabo con el propósito de identificar y documentar los sistemas de archivos, servicios y usuarios presentes en el sistema, así como para evaluar posibles vulnerabilidades o configuraciones que puedan representar riesgos de seguridad.

El firmware analizado corresponde a una imagen extraída de una camara y de una boombilla, la cual fue examinada utilizando herramientas especializadas como binwalk. Estas herramientas permitieron descomponer la imagen del firmware, extraer los sistemas de archivos y analizar su contenido de manera sistemática.

2. Metodología

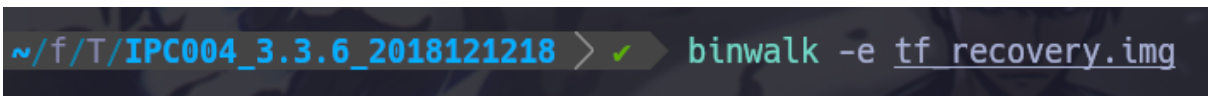
Análisis de la bombilla IoT:

- En el caso de la bombilla IoT, se intentó extraer y analizar el firmware utilizando las mismas herramientas y técnicas. Sin embargo, no se pudo obtener información relevante debido a las restricciones impuestas por el desarrollador.

El análisis forense de la cámara se realizó siguiendo una metodología estructurada que incluyó las siguientes etapas:

1. **Extracción del firmware:**
    - Se utilizó la herramienta binwalk para descomponer la imagen del firmware (tf_recovery.img) y extraer los sistemas de archivos contenidos en ella. Binwalk es una herramienta ampliamente utilizada en análisis forense para identificar y extraer archivos incrustados en imágenes de firmware.
    - El comando utilizado fue: binwalk -e tf_recovery.img

    Este comando permitió identificar los sistemas de archivos presentes en la imagen y extraerlos para su posterior análisis



2. **Identificación de sistemas de archivos:**

    - Una vez extraídos los sistemas de archivos, se procedió a identificar los sistemas de archivos presentes en la imagen, utilizando tanto binwalk como la inspección manual de los archivos extraídos.

3. **Análisis de servicios:**
    - Se examinaron los scripts de inicio y los archivos de configuración para identificar los servicios en ejecución. Esto incluyó la revisión de archivos en la ruta squashfs-root/etc/init.d/.

4. **Análisis de usuarios:**
    - Se revisó el archivo /etc/passwd para listar los usuarios del sistema y sus configuraciones. Este archivo contiene información sobre los usuarios del sistema, incluyendo sus shells y rutas de inicio.

5. **Documentación de hallazgos:**
    - Todos los hallazgos se documentaron de manera detallada, incluyendo capturas de pantalla y comandos utilizados.

3. Análisis del Firmware de la Bombilla IoT

3.1. Información Obtenida

En el caso de la bombilla IoT, no se pudo extraer información relevante del firmware. El desarrollador ha implementado medidas para evitar que se pueda acceder al código o a los sistemas de archivos del dispositivo. Esto incluye posiblemente el uso de técnicas como:

- **Cifrado del firmware:** El firmware puede estar cifrado, lo que impide su extracción y análisis.
- **Protección contra desensamblado:** El desarrollador puede haber utilizado técnicas para dificultar la ingeniería inversa del firmware.

3.2. Implicaciones para el Análisis Forense

Esta situación supone un desafío significativo para el análisis forense, ya que:

- **Limitación de información**: Sin acceso al firmware, no es posible identificar vulnerabilidades, servicios en ejecución o usuarios del sistema.

- **Dificultad para detectar malware:** Si el dispositivo está comprometido, no se puede analizar el firmware para detectar la presencia de malware o actividades sospechosas.

- **Falta de transparencia:** La imposibilidad de analizar el firmware puede indicar que el desarrollador no está siguiendo buenas prácticas de seguridad, lo que podría ser un riesgo para los usuarios.

3.3. Conclusión

La imposibilidad de analizar el firmware de la bombilla IoT debido a las restricciones impuestas por el desarrollador dificulta la identificación de vulnerabilidades y la realización de un análisis forense completo. Esto resalta la importancia de que los desarrolladores de dispositivos IoT sigan prácticas de seguridad transparentes y permitan el análisis de sus firmwares.

4. Análisis de Sistemas de Archivos de la Cámara

4.1. Analisis Sistemas de Archivos Identificados

En el firmware analizado se identificaron dos sistemas de archivos principales:

1. **Squashfs:**
    - Descripción: Squashfs es un sistema de archivos comprimido de solo lectura, optimizado para almacenar archivos de manera eficiente en términos de espacio. Es comúnmente utilizado en dispositivos embebidos para almacenar el sistema operativo y los archivos esenciales que no requieren modificaciones frecuentes.
    - Uso en el firmware: En este caso, Squashfs se utiliza para almacenar la parte comprimida del firmware, que incluye el sistema operativo y los archivos esenciales del dispositivo.

2. **JFFS2 (Journaling Flash File System 2):**
    - Descripción: JFFS2 es un sistema de archivos diseñado específicamente para dispositivos de almacenamiento flash. A diferencia de Squashfs, JFFS2 permite operaciones de lectura y escritura, lo que lo hace ideal para almacenar datos que cambian con el tiempo, como configuraciones del dispositivo o registros de actividad.
    - Uso en el firmware: JFFS2 se utiliza en este firmware para gestionar el almacenamiento de la memoria flash, donde se guardan datos que pueden modificarse durante el funcionamiento del dispositivo.

Ambos sistemas de archivos están presentes en la imagen del firmware extraída, con Squashfs utilizado para la parte comprimida y JFFS2 gestionando el almacenamiento de la memoria flash.

4.2. Conclusión

La combinación de Squashfs y JFFS2 es común en dispositivos embebidos, ya que permite separar el sistema operativo y los archivos esenciales (que no cambian) de los datos que pueden modificarse durante el uso del dispositivo. Esta separación mejora la eficiencia y la durabilidad del almacenamiento flash.

5. Análisis de Servicios de la Cámara

5.1. Servicios Identificados

En el análisis del sistema de archivos extraído, se identificaron los siguientes servicios:

1. **mdev:**
    - **Descripción:** mdev es un gestor de dispositivos en sistemas Linux, utilizado para manejar la detección y configuración de dispositivos cuando son añadidos o eliminados. Es común en entornos de arranque ligero y sistemas embebidos, como los que se encuentran en dispositivos móviles y de IoT.
    - **Función en el firmware:** Este servicio se encarga de gestionar los dispositivos conectados al sistema, asegurando que estén correctamente configurados y disponibles para su uso.

2. **gm:**
    - **Descripción:** Aunque el nombre exacto del servicio no se especifica en el script, es probable que gm esté relacionado con la gestión de red o dispositivos. En sistemas embebidos, este tipo de servicio suele ser responsable de la configuración y el manejo de la conectividad de red.
    - **Función en el firmware:** Posiblemente gestiona la configuración de red del dispositivo, asegurando que esté correctamente conectado a la red y pueda comunicarse con otros dispositivos.

3. **miio_avstreamer:**
    - **Descripción:** Este servicio está relacionado con la transmisión de audio y video. Es común en dispositivos que requieren la capacidad de transmitir contenido multimedia, como cámaras de seguridad o dispositivos de entretenimiento.
    - **Función en el firmware:** Se encarga de gestionar la transmisión de audio y video, lo que sugiere que el dispositivo puede tener capacidades multimedia.

4. **miio_client:**

- **Descripción:** Este servicio es un cliente que interactúa con otros dispositivos en la red, posiblemente relacionado con la plataforma MIIO (Xiaomi Internet of Things). Permite la comunicación entre dispositivos inteligentes y su control.
- **Función en el firmware:** Facilita la comunicación del dispositivo con otros dispositivos IoT, permitiendo su integración en un ecosistema de dispositivos inteligentes.

5. **miot_devicekit:**
   - **Descripción:** Este servicio es parte de un kit de desarrollo para dispositivos IoT de Xiaomi. Proporciona herramientas y bibliotecas para el desarrollo y la gestión de dispositivos en la red.
   - **Función en el firmware:** Proporciona las herramientas necesarias para gestionar el dispositivo dentro de un entorno IoT, facilitando su configuración y mantenimiento.

Estos servicios se inician en el script rcS de la ruta squashfs-root/etc/init.d/, donde se pueden ver las llamadas a cada uno, indicando su importancia en el proceso de arranque del sistema.

Comando usado:
grep -r "start" squashfs-root/etc/init.d

5.2. Conclusión

Los servicios identificados son críticos para el funcionamiento del dispositivo, especialmente en un entorno IoT. Servicios como miio_client y miot_devicekit sugieren que el dispositivo está diseñado para integrarse en un ecosistema de dispositivos inteligentes, mientras que mdev y gm aseguran la correcta configuración y conectividad del dispositivo.

6. Análisis de Usuarios

6.1. Usuarios Identificados

El análisis del archivo /etc/passwd dentro del sistema de archivos extraído reveló la siguiente lista de usuarios:

1. **root:**
    - Descripción: El usuario principal del sistema con privilegios totales. Tiene acceso completo al sistema y se encuentra en la ruta /root, utilizando la shell /bin/sh.
    - Riesgo: Este usuario es crítico, ya que un acceso no autorizado a la cuenta de root podría comprometer todo el sistema.

2. **daemon:**
    - Descripción: Usuario designado para ejecutar procesos en segundo plano que no requieren acceso a la cuenta de root. Su shell está deshabilitada (/bin/false), lo que significa que no puede iniciar sesión interactivamente.
    - Riesgo: Bajo, ya que no permite acceso interactivo.

3. **bin:**
    - Descripción: Usuario asociado a la gestión de archivos de binarios del sistema. Su shell está deshabilitada.
    - Riesgo: Bajo.

4. **sys:**
    - Descripción: Usuario del sistema utilizado para tareas administrativas y de mantenimiento. Su shell está deshabilitada.
    - Riesgo: Bajo.

5. **sync:**
    - Descripción: Usuario utilizado para permitir la sincronización de datos en el sistema. Su shell está configurada para ejecutar el comando de sincronización.
    - Riesgo: Bajo.

6. **mail:**
    - Descripción: Usuario responsable de gestionar el correo electrónico del sistema. Su shell está deshabilitada.
    - Riesgo: Bajo.

7. **www-data:**

- Descripción: Usuario asociado a servidores web, como Apache. Ejecuta procesos relacionados con la entrega de contenido web. Su shell está deshabilitada.
- Riesgo: Bajo.

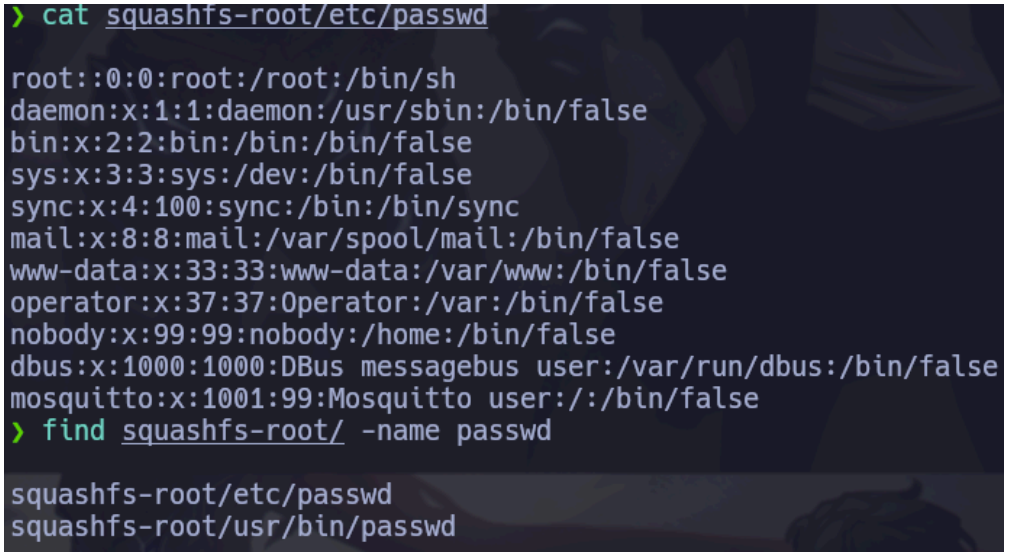
8. **operator:**
   - Descripción: Usuario con privilegios adicionales para realizar tareas de mantenimiento en el sistema. Su shell está deshabilitada.
   - Riesgo: Moderado, ya que tiene privilegios adicionales.

9. **nobody:**
   - Descripción: Usuario sin privilegios utilizado para ejecutar procesos que no requieren permisos específicos. Su shell está deshabilitada.
   - Riesgo: Bajo.

10. **dbus:**
    - Descripción: Usuario designado para manejar el servicio de mensajería D-Bus, que permite la comunicación entre aplicaciones. Su shell está deshabilitada.
    - Riesgo: Bajo.

11. **mosquitto:**
    - Descripción: Usuario relacionado con el broker MQTT Mosquitto, utilizado para la gestión de mensajes en aplicaciones IoT. Su shell está deshabilitada.
    - Riesgo: Bajo.

La información sobre estos usuarios se extrae mediante el comando “cat squashfs-root/etc/passwd” y “find squash-root/ - name passwd”, que contiene las credenciales y configuraciones básicas de los usuarios del sistema.



6.2. Conclusión

Los usuarios identificados son típicos de sistemas Linux, con el usuario root siendo el más crítico debido a sus privilegios totales. La mayoría de los usuarios tienen shells deshabilitadas, lo que reduce el riesgo de accesos no autorizados. Sin embargo, es importante asegurar que el usuario root esté protegido con una contraseña fuerte y que se monitorice su actividad.

7. Tipo de Análisis Realizado

7.1. ¿Cómo se llama este tipo de análisis?

El tipo de análisis realizado se conoce como análisis forense digital. Este enfoque se utiliza para examinar dispositivos y sistemas de manera sistemática, con el objetivo de identificar, preservar y analizar datos relevantes relacionados con incidentes de seguridad o mal funcionamiento. En este caso específico, el análisis se centró en la extracción y evaluación de la imagen del firmware (tf_recovery.img), utilizando herramientas como binwalk y Firmwalker para descubrir y examinar el contenido de los sistemas de archivos, así como para identificar posibles servicios y usuarios en el entorno del sistema.

7.2. Ubicación en el Informe

Esta sección se ubica después del Análisis de Usuarios y antes de las Conclusiones Generales, ya que proporciona una descripción del tipo de análisis realizado y su importancia en el contexto del trabajo. Es esencial destacar que el análisis forense digital es una metodología clave en la ciberseguridad, ya que permite:

- Identificar vulnerabilidades: A través del análisis de los sistemas de archivos, servicios y usuarios, se pueden detectar posibles puntos débiles en la seguridad del dispositivo.
- Preservar evidencias: El análisis forense asegura que los datos sean extraídos y documentados de manera que puedan ser utilizados como evidencia en caso de incidentes de seguridad.
- Comprender el funcionamiento del sistema: Este tipo de análisis permite entender cómo está estructurado el firmware, qué servicios están en ejecución y qué usuarios tienen acceso al sistema.

7.3. Herramientas Utilizadas

- Binwalk: Herramienta utilizada para extraer y analizar la imagen del firmware. Permite identificar y descomponer los sistemas de archivos incrustados en la imagen.

7.4. Conclusión

El análisis forense digital es una metodología esencial en la ciberseguridad, especialmente cuando se trata de dispositivos embebidos y sistemas IoT. En este caso, el análisis permitió no solo extraer y examinar el firmware, sino también identificar los sistemas de archivos, servicios y usuarios presentes en el dispositivo. Este tipo de análisis es fundamental para garantizar la seguridad y el correcto funcionamiento de los dispositivos, así como para detectar y mitigar posibles vulnerabilidades.

8. Conclusiones Generales

Cámara:
El análisis forense del firmware permitió identificar los sistemas de archivos, servicios y usuarios presentes en el dispositivo. La combinación de Squashfs y JFFS2 es adecuada para dispositivos embebidos, ya que separa los archivos esenciales de los datos modificables. Los servicios identificados sugieren que el dispositivo está diseñado para funcionar en un entorno IoT, con capacidades de transmisión de audio y video.

Los usuarios del sistema son típicos de un entorno Linux, con el usuario root siendo el más crítico. La mayoría de los usuarios tienen shells deshabilitadas, lo que reduce el riesgo de accesos no autorizados.

El tipo de análisis realizado, análisis forense digital, es esencial para entender la estructura y el funcionamiento del firmware, así como para identificar posibles vulnerabilidades y recopilar evidencias en caso de incidentes de seguridad.

Bombilla IoT:
No se pudo extraer información relevante del firmware debido a las restricciones impuestas por el desarrollador. Esto dificulta el análisis forense y resalta la importancia de la transparencia en la seguridad de los dispositivos IoT.

9. Recomendaciones

1. **Protección del usuario root:** Asegurar que el usuario root esté protegido con una contraseña fuerte y que se monitorice su actividad.

2. **Auditoría de servicios:** Revisar los servicios en ejecución para detectar posibles vulnerabilidades y asegurar que solo los servicios necesarios estén activos.

3. **Actualización del firmware:** Mantener el firmware actualizado para corregir posibles vulnerabilidades y mejorar la seguridad del dispositivo.

4. **Monitorización de actividad:** Implementar herramientas de monitorización para detectar actividades sospechosas o accesos no autorizados.

5. **Transparencia en el desarrollo:** Los desarrolladores de dispositivos IoT deben permitir el análisis de sus firmwares para garantizar la seguridad de los usuarios

10. Identificación y Custodia de la Evidencia

Para garantizar la integridad de los firmwares analizados y demostrar que no han sido modificados durante el proceso de análisis, se calcularon los hashes MD5 y SHA-256 de los archivos de firmware. Estos hashes actúan como "huellas digitales" únicas de los archivos, permitiendo verificar su autenticidad en cualquier momento.

Firmware de la Cámara:

- **Nombre del archivo:** tf_recovery.img
- **Hash MD5:** 659fe1b7b94046ff58ba2649e13f198c
- **Hash SHA-256:** eb94f19a9f7863eb8c4523e435a06dee365d236b8ea549ef1529ccc9bb323dcb

Firmware de la Bombilla:

- **Nombre del archivo:** ZW098_LED_Bulb_G5_AU_A_V1_07.ex_
- **Hash MD5:** a6f8471ad1df1e0319e0e78403ed9f4e
- **Hash SHA-256:** ccf777d33fdb868e1dc9c4d71dfb44e1bad02b3e8cb0245983547c3777f0dd15

Proceso de Verificación:

1. **Cálculo de los hashes:** Los hashes se calcularon utilizando las herramientas md5sum y sha256sum en un entorno Linux.
2. **Verificación de integridad:** Para asegurar que los firmwares no han sido alterados, se recomienda recalcular los hashes y compararlos con los valores originales.

Importancia de la Custodia de la Evidencia:

- **Integridad:** Los hashes garantizan que los firmwares no han sido modificados.
- **Transparencia:** Permite la verificación por parte de otros investigadores.
- **Cumplimiento legal:** Esencial para garantizar que los datos puedan ser admitidos como prueba en un proceso legal.

11. Certificación del Análisis

Este informe certifica que el análisis forense digital aquí descrito fue realizado siguiendo las mejores prácticas de la disciplina, utilizando herramientas reconocidas en el ámbito forense y bajo un entorno controlado. Las evidencias fueron manejadas conforme a los estándares legales y técnicos, asegurando la cadena de custodia y la integridad de los datos obtenidos.

Datos del Investigador:

- **Nombre**: Xinwei
- **Apellidos**: Wu
- **DNI/NIE**: U8766008F
- **Teléfono**: 600 000 000
- **Correo electrónico**: xinwei.wu@forense.digital.es
- **Fecha de elaboración del informe**: 12 de marzo de 2025

Firma:
(Firma del investigador o equipo forense)
Xinwei Wu

10. Anexos

- **Comandos utilizados:**
    - binwalk -e tf_recovery.img
    - grep -r "start" squashfs-root/etc/init.d
    - cat squashfs-root/etc/passwd