Grado en Ingeniería Informática

Trabajo de Final de Grado

Estudio y análisis de vulnerabilidades en dispositivos IoT

Realizado por:
Ahmed Ratleh Masmas

Supervisado por:
Manuel Ginés Rodríguez
Tutorizado por:
Manuel Francisco Dolz Zaragozá

Fecha de lectura: 15 de Septiembre de 2022
Curso académico 2021/2022

Resumen

Este proyecto presenta el desarrollo de una metodología para realizar auditorías a dispositivos hardware IoT. Esta metodología permitirá descubrir y analizar las vulnerabilidades del dispositivo auditado. A lo largo del proyecto se definirán los requisitos necesarios para la realización de esta metodología. Se diseñará el proceso de auditoría en base a los requisitos establecidos. Finalmente, se aplica la metodología desarrollada a tres dispositivos: una cerradura inteligente, una aplicación móvil y un dispositivo PLC.

Palabras clave

IoT, hardware hacking, pentesting, vulnerabilidades, vector de ataque, WiFi, NFC, firmware, puertos, protocolos, auditoría

Keywords

IoT, hardware hacking, pentesting, vulnerabilities, attack vector, WiFi, NFC, firmware, ports, protocols, audit

Agradecimientos

Mis primeros agradecimientos son para mis padres y mis hermanas por apoyarme en cada momento de la carrera, desde la espontánea decisión de elegirla hasta el final de ella, ayudándome en cada momento que lo necesitaba. Y también al resto de mi familia, que me apoyaban constantemente desde la distancia pese a estar en tiempos de guerra.

También quisiera agradecer a todos los compañeros de clase con los que he establecido una buena amistad a lo largo de todas las asignaturas de la carrera, y sobretodo al grupo de amigos al que llegué a pertenecer, ya que gracias a ellos conseguí centrarme y ponerme las pilas para sacarlo todo adelante, todo entre risas.

Asimismo, quiero agradecer a muchos de mis amigos y personas muy cercanas que he conocido a través de Internet por también apoyarme y ayudarme con el curso de la carrera, superando malas épocas y temporadas y distrayéndome y animándome siempre que lo necesitara, y sobretodo estando ahí por mí.

Además agradecer a todos los profesores y tutores que me han hecho llegar hasta este punto, todos los que me apoyaron, ayudaron y confiaron en mí en cada una de mis épocas de estudiante.

Y finalmente, agradecer a mi tutor Manuel Dolz por toda la paciencia que ha tenido a lo largo de la producción del trabajo de final de grado; a mi supervisor Manuel Ginés, por hacer posible una estancia en prácticas tan inusual pero a la vez tan genial, además de aconsejarme sobre qué planes y proyectos futuros tomar y ayudarme siempre que lo necesitaba; y a la empresa que hizo el proyecto posible, Soluciones Cuatroochenta, Sofistic.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Este documento constituye el **trabajo de final de grado** del alumno **Ahmed Ratleh Masmas**. En este capítulo se dará una pequeña **introducción** acerca del proyecto, de su contexto y de la empresa en la que se ha desarrollado.

1.1. Contexto y motivación del proyecto

El proyecto llevará a cabo el **desarrollo de una metodología de auditoría a dispositivos IoT**, (del inglés *Internet of Things*, Internet de las Cosas), donde se estudiarán e investigarán todas las vulnerabilidades de estos dispositivos. Este proyecto ha sido desarrollado en la empresa **Soluciones Cuatroochenta S.A.**, en su correspondiente sector de **ciberseguridad** (**Sofistic**).

La principal motivación del proyecto es analizar y descubrir las vulnerabilidades de los dispositivos hardware IoT y desarrollar una metodología genérica. Para ello, se realizarán **auditorías de seguridad** de algunos de estos dispositivos. Se investigará desde un amplio **vector de ataque** las posibles vulnerabilidades de los mismos, incluyendo los protocolos de comunicación, los puertos que utilizan los dispositivos o en el diseño físico los dispositivos. Se realizan las auditorías en base a ese vector. Por último, se redacta un informe final orientado a la entidad propietaria del dispositivo.

1.1.1. Sobre la empresa

La empresa Soluciones Cuatroochenta S.A. [10] (en adelante **Cuatroochenta**), localizada en la **Universitat Jaume I** de Castellón de la Plana, se dedica a desarrollar **soluciones digitales cloud y ciberseguridad**, teniendo como objetivo reducir la distancia entre lo que una empresa cliente realiza y lo que es capaz de realizar, garantizando la ciberseguridad de esta. Su logo se ve en la Figura 1.1.

La empresa encargada de la **ciberseguridad** de los clientes es **Sofistic** [49], propiedad de Cuatroochenta. Su logotipo se puede ver en la Figura 1.2. Sofistic se encuentra en diversos países del continente americano y en España. Se ha dedicado durante años a ofrecer una correcta

Figura 1.1: Logo de Cuatroochenta.

seguridad mediante sus auditorías, entre otros servicios que ofrece. Algunas auditorías están basadas en pruebas de penetración (**pentesting**), auditorías acerca de la red WiFi o un test de carga de red. Respecto a los demás servicios, pueden realizar consultorías de seguridad, investigación y análisis digital y forense de la empresa y otros servicios gestionados, como la monitorización de registros de datos (**logs**) o la administración y supervisión de los sistemas de la empresa cliente.

Figura 1.2: Logo de Sofistic.

El negocio de la empresa se basa en los clientes que **solicitan ser auditados** por Sofistic para garantizar su propia seguridad, donde se puede solicitar un presupuesto según las necesidades del cliente.

El lugar de trabajo se sitúa en el edificio **Espaitec 2** en la Universitat Jaume I de Castellón de la Plana, en la cuarta planta. La Figura 1.3 muestra una imagen del edificio en cuestión. Allí se encuentran las oficinas de Cuatroochenta con sus respectivos departamentos para las empresas que esta contiene, como Sofistic.

Figura 1.3: Edificio Espaitec 2 en la UJI.

1.1.2. Sobre la auditoría de dispositivos IoT

Cada año que pasa se desarrolla y se genera una gran variedad de **dispositivos** que se conectan a Internet. Estos se denominan dispositivos **IoT** (del inglés *Internet of Things*, Internet de las Cosas). Estos dispositivos pueden ser, de manera general, cualquier **dispositivo con conexión a Internet**, **Bluetooth** u **otros medios inalámbricos**, como bombillas, enchufes o neveras inteligentes. Desafortunadamente, debido a la producción en masa de estos dispositivos y la rapidez de los fabricantes en ser los primeros en desarrollar un nuevo dispositivo IoT, la seguridad informática del dispositivo desarrollado no siempre se tiene en cuenta como es debido. Es **necesario**, por tanto, garantizar la **máxima seguridad** de estos dispositivos, ya que están presentes de las maneras más convencionales en las casas, y cualquier vulnerabilidad de estos puede ser **crítico** para la **integridad** y la **seguridad** del **usuario** que los posee.

Para poder garantizar la **máxima seguridad informática** de los dispositivos, se realizan **auditorías** sobre los dispositivos para descubrir todas las **vulnerabilidades** posibles que puedan suponer un riesgo para el usuario. Para auditar estos dispositivos es necesario seguir un protocolo, proceso o **metodología** para obtener la máxima cantidad de vías donde se puedan encontrar estas vulnerabilidades, sea con o sin acceso físico a este tipo de dispositivos.

Estas auditorías se pueden realizar de manera independiente o desde una empresa. La empresa Sofistic se dedica, principalmente, a la ciberseguridad de las empresas realizando este tipo de auditorías, por tanto tienen una amplia experiencia en este sector. No obstante, el proceso para realizar auditorías sobre dispositivos IoT puede variar.

1.1.3. Sobre las tareas a realizar

Las tareas realizadas en la empresa se basan en **desarrollar** e **implementar** una metodología de **auditoría de dispositivos IoT**. Para ello, **primero** se realizó una **formación** acerca de cómo identificar protocolos y cómo poder auditarlos. En esta formación se utilizaron una **cerradura inteligente** y una **aplicación móvil**. Durante este proceso había que aprender nuevos protocolos, cómo se pueden auditar y vulnerar, cómo descubrir que lo descubierto puede ser una vulnerabilidad y cómo de crítica es y cómo realizar un informe de la auditoría completa orientado al cliente. Una vez aprendidos los conceptos básicos y cómo auditar correctamente un dispositivo, la **tarea final** era **auditar por cuenta propia** un dispositivo sin previa auditoría por parte de la empresa, respetando el proceso de auditoría diseñado. Este dispositivo sería un PLC (del inglés *Power Line Communications*, Comunicaciones vía Red Eléctrica).

1.1.4. Alcance del proyecto

El alcance de este proyecto consiste en **desarrollar una metodología** para auditar cualquier dispositivo IoT hardware. Esto se hará **auditando** un dispositivo, investigando sus protocolos y puertos de conexión, y procurando entender cómo funciona internamente a partir de su estructura de comunicación y diseño de los circuitos. También se investiga si el dispositivo dispone de una aplicación web de configuración o una aplicación móvil para auditarla de manera adecuada, se puede desensamblar el firmware del dispositivo extrayéndolo desde sus memorias (o descargándolo de Internet). También se pueden interceptar las conexiones inalámbricas del dispositivo para investigar si su protocolo de comunicaciones es o no seguro. Una vez **descu-**

biertas las posibles **vulnerabilidades** que pudiera tener, **se ofrecerán** las **soluciones** a estas para, finalmente, redactar un **informe final** que resuma todos los resultados de la auditoría incluyendo estas soluciones. Una vulnerabilidad puede ser, por ejemplo, almacenar una contraseña sin cifrar; su solución sería almacenar la contraseña cifrada.

1.2. Estructura de la memoria

Esta memoria se divide en diversos capítulos:

- En el **Capítulo 2** de **Planificación del proyecto** se explica la metodología empleada en la estancia en prácticas para llevar a cabo el proyecto, junto con sus objetivos y tareas. También se incluyen estimaciones de los recursos utilizados en este (humanos, hardware y software).
- Seguido de ello, en el **Capítulo 3** de **Análisis y diseño del procedimiento de la auditoría** se definen los requisitos del proyecto. Estos son los requisitos de usuario y los de la metodología de auditoría. También se describe el proceso de auditoría mediante diagramas de casos de uso y de flujo para una mejor explicación. Se incluye una descripción sobre cómo se debería aplicar esta metodología. Finalmente se encuentra una matriz de trazabilidad de los requisitos del proyecto.
- En el **Capítulo 4** de **Aplicación de la metodología de auditoría** se aplica el proceso de auditoría aprendido en el capítulo anterior. Se aplicará para tres elementos: una cerradura inteligente, una aplicación móvil y un dispositivo PLC.
- Finalmente, en el **Capítulo 5** de **Conclusiones y trabajo futuro** se da una conclusión al proyecto.

Debido a la **confidencialidad** de los datos, según lo manifestado por la empresa, y para mantener este documento público, se ha decidido **censurar todos los nombres** de todo aquello auditado. En su lugar, se escribirán sinónimos del tipo de dispositivo que son.

Capítulo 2

Planificación del proyecto

En este capítulo se detallarán la metodología y la planificación del proyecto junto con los objetivos de este. De manera adicional, se desarrolla un plan de contingencia para todas las desviaciones que puedan suceder. Por último, se añaden los recursos utilizados para este proyecto y una estimación de sus costes.

2.1. Metodología

La **metodología** del proyecto formativo es de tipo **predictiva** [41]. Esta metodología se basa en la predicción del trabajo a realizar durante el proyecto. Se aplica esta metodología ya que es la más apta para la naturaleza de un proyecto de análisis e investigación. Debido a que se basa en la predicción del trabajo futuro, pueden ocurrir riesgos durante esta planificación que no permitan el cumplimiento de esta. Por ello, se debe desarrollar un control de riesgos o **plan de contingencia** para poder prevenir cualquier problema que pueda surgir en el futuro, siempre respetando que sea lo más cercano posible a la planificación inicial.

2.2. Objetivos del proyecto

2.2.1. Objetivos

El principal objetivo de este proyecto es desarrollar una metodología para auditar dispositivos IoT, sea cual sea el propósito de este. Para ello se deben analizar y descubrir las vulnerabilidades del dispositivo hardware e investigar el diseño y funcionamiento de su circuito. Para llevar a cabo este desarrollo, se desglosan los siguientes **subobjetivos**:

- Identificar el propósito del dispositivo.
- Conocer cómo funciona internamente un dispositivo hardware IoT, identificando los componentes del circuito y averiguando las conexiones entre ellos.
- Interactuar con las diferentes tecnologías de comunicación inalámbricas (WiFi, NFC, etc.)

e identificar las posibles vulnerabilidades que puedan contener a nivel de comunicación y de protección de la información.

- Interactuar con los diferentes sistemas que pueda incorporar (aplicaciones móviles, sensores, etc.) e identificar posibles vulnerabilidades que afecten a la seguridad del usuario.
- Desensamblar el firmware de un dispositivo para averiguar lo máximo posible su funcionamiento y su seguridad.
- Ofrecer posibles soluciones a las vulnerabilidades encontradas.
- Redactar un informe formal que englobe los resultados de la auditoría.

Ya realizados los subobjetivos, se establecen un orden y unos procesos más concretos de cada uno de ellos para llevar a cabo el diseño y desarrollo de la metodología.

2.3. Alcance del proyecto

El alcance del proyecto mencionado en el **Capítulo 1** de **Introducción** se puede desglosar en tres alcances: funcional, organizativo y tecnológico.

2.3.1. Alcance funcional

El alcance funcional de este proyecto tiene como finalidad ofrecer una nueva metodología de auditoría para la empresa. Esta se desarrolla mediante auditorías de seguridad sobre varios dispositivos IoT, para englobar las máximas vías y posibilidades de encontrar vulnerabilidades en cualquier otro dispositivo de este tipo.

2.3.2. Alcance organizativo

Respecto al alcance organizativo, el proyecto se realiza en el departamento de ciberseguridad de la empresa, la cual tiene una amplia experiencia en auditorías de seguridad, principalmente web, entre otros servicios. Esta se encargará de garantizar la seguridad del cliente mediante esas auditorías. En cuanto al alcance interno a la empresa, este proyecto recopila una metodología de realizar auditorías desarrollada por el alumno.

2.3.3. Alcance tecnológico

En relación al alcance tecnológico o informático, se utilizarán los recursos software y hardware que se documenten en la **Sección 2.4** de **Estimación de recursos y costes del proyecto**.

2.4. Estimación de recursos y costes del proyecto

2.4.1. Recursos software y su coste

Se han requerido de ciertos **recursos software** para poder realizar las auditorías. Todo el **software** utilizado es **libre** y **gratuito**, por tanto no tiene ningún coste adicional. Estos programas son los siguientes:

- Aplicaciones móviles: nRF Connect, NFC Tools, MIFARE Classic Tools
- Aplicaciones de ordenador: Wireshark, ubertooth, GATTacker, hcitool, gattool, GHidra, OWASP Zaproxy, Logic, Xgpro, Serial Monitor Tool, nmap
- Máquinas virtuales (Oracle Virtual Box): Ubuntu 20 y Windows 10
- Aplicaciones para la empresa: Repositorio de aplicaciones de Microsoft (Teams, Authenticator, etc.)

Lo **más común** es utilizar estos programas en máquinas basadas en **Linux** para evitar problemas de compatibilidades o de funcionamiento. Cabe destacar que **no siempre** funcionan mejor en Linux.

2.4.2. Recursos hardware y su coste

Durante el proyecto se han requerido de diversos **materiales hardware** para realizar las auditorías, que se muestran en la Tabla 2.1. La mayoría de precios mostrados en la tabla, por ejemplo los de los dispositivos, son una media aproximada debido a que existe una gran variedad de parecidos con distintos precios. Lo natural es que un cliente solicite la auditoría de su dispositivo a la empresa, por tanto **no** se debería **añadir su coste** a los costes hardware del proyecto. Sin embargo, al ser un periodo de **entrenamiento** y de **desarrollo de una metodología de auditoría**, estos productos fueron adquiridos por la empresa con ese fin.

Recurso	**Coste (€)**	**Tiempo de amortización en meses**	**Coste prorrateado a 300 horas (€)**
Ordenador de empresa (Macbook Pro 2015)	540	60	3,7
Pantalla 22”	150	60	1,03
Docking station	80	24	1,37
Soporte para portátil con ventilador	20	24	0,35
Teclado y ratón	25	24	0,35
Dispositivo móvil (Android 9 o superior con buen zoom)	500	30	6,85
Enrutador antiguo de segunda mano	20	12	0,65
Cerradura inteligente	200	24	3,43
Dispositivo PLC	60	48	0,52
Kit de herramientas y destornilladores	23	120	0,08
Cables tipo dupont	3	30	0,05
Multímetro	15	60	0,11
Estación de soldadura	95	60	0,66
Estaño y flux para soldar	4	12	0,14
Alfombrilla para soldar	15	48	0,13
Adaptador UART-USB	8	120	0,03
USB Bluetooth dongle	3	60	0,03
Lector de memorias EPROM con adaptadores	110	120	0,38
Analizador lógico	20	120	0,07
Total			**19,93**

Tabla 2.1: Tabla de recursos hardware y sus costes.

Todos los recursos se han utilizados mínimo una vez durante la estancia en prácticas. Es decir, todos han sido necesarios para al menos concluir con las auditorías realizadas. La gran mayoría de estos materiales fueron proporcionados por la empresa.

2.4.3. Recursos humanos e infraestructurales y su coste

Respecto a los **recursos humanos** se encuentran los gastos de contratación de un **empleado junior** y su **supervisor**. El puesto de empleo será de **analista de ciberseguridad junior**, y su supervisor será un **experto en ciberseguridad**. En relación al puesto de analista de ciberseguridad junior no hay mucha información. En España, se estima con baja confianza que el salario en jornada completa sea de **25.000 €**, según [27]. En otros países como Estados Unidos, el salario más bajo encontrado, en jornada completa, es de **23.000 $**. Mientras que el de un experto en ciberseguridad (este sí dispone de más datos y más fiables) oscila entre **30.000 €** y **60.000 €** brutos anuales, dando una media de **45.000 €**, también en jornada completa. Partiendo de los salarios anuales brutos de 25.000€para el empleado junior y de 45.000 €para el supervisor, se hacen los respectivos cálculos en los dos salarios para computarlos a **300 horas** y **100 horas**, respectivamente. Los costes de contratación por parte de la empresa oscilan entre el 31 % y el 33 % del salario bruto, según un artículo del banco **Laboral Kutxa** [9] y un artículo del blog **Infoautónomos** [32], basado en la **Ley 42/2006, de 28 de diciembre, de Presupuestos Generales del Estado para el año 2007** [5]. Por tanto se tomará un término

intermedio de 32 %. Todos estos datos se recopilan en la Tabla 2.2.

Recurso	**Cantidad de horas (h)**	**Coste por hora (€/h)**	**Total (€)**
Empleado junior	300	13,02	3.906
Supervisor	100	23,44	2.344
Total (€)			**6.250**

Tabla 2.2: Tabla de recursos humanos y sus costes.

Los gastos **infraestructurales** se relacionan con el gasto invertido en la electricidad, el Internet y el alquiler de la zona de las oficinas. En la Tabla 2.4 se encuentran resumidos estos gastos. Todos aquellos costes que se tengan en cuenta mensualmente se han prorrateado a 300 horas, teniendo en cuenta que un mes dispone de 730 horas.

Los gastos de **alquiler** se pueden calcular mediante la **calculadora del espaitec** [19]. Se sabe que es una **empresa** de colaboración **física**, **en crecimiento** y que abarca la cuarta planta del edificio Espaitec 2. Este edificio [20] reparte módulos de 230 metros cuadrados como oficinas. En la cuarta planta del edificio se encuentran 5 zonas de acceso, por tanto se intuye que dispone de 5 módulos, dando como resultado 1.150 metros cuadrados. Si se utiliza otra herramienta [6] para medir el área de las oficinas, se obtiene aproximadamente 1.053 metros cuadrados. Así que se tomará como base **1.100 metros cuadrados** para los cálculos de la página citada anteriormente. Como resultado se obtienen, aproximadamente, **11.597 €** al mes. Prorrateando este coste a las 300 horas del proyecto y, manteniendo la relación de que cada mes tiene 730 horas, el coste es de, aproximadamente, **4.765,89 €**. Los costes de la **electricidad** y el **Internet no están incluidos**, pero se garantiza limpieza y seguridad durante las 24h.

Respecto al coste de electricidad, se puede medir el consumo necesario con los dispositivos utilizados. En la Tabla 2.3 se identifica el coste prorrateado de estos dispositivos.

Recurso	**Cantidad de horas (h)**	**Consumo por hora (Wh)**	**Total (kWh)**
Ordenador	300	85	25,5
Pantalla	300	23	6,9
Enrutador antiguo	50	6	0,3
Cerradura inteligente	50	5	0,25
Dispositivo PLC	150	6	0,9
Estación de soldadura	5	60	0,3
Total (kWh)			**34,15**

Tabla 2.3: Tabla de consumo de los dispositivos utilizados.

Teniendo en cuenta que el total de consumo es de **34,15 kWh** (kilovatios hora) y que, actualmente, el precio de kilovatio por hora es de **0,2460 €/kWh**, se dispone de un coste eléctrico total del proyecto de **8,40 €**.

En cuanto al consumo de Internet, se puede utilizar un plan de empresa. Tomando como ejemplo el siguiente plan [37], teniendo en cuenta su fecha de visita, incluyendo el IVA (Impuesto sobre el Valor Añadido) y sin aplicar ningún precio de oferta ni promoción, para un plan de **1 Gbps** de velocidad simétrica de Internet el coste es de **57 €** al mes. Prorrateando este coste a las 300 horas del proyecto se obtiene un total de **23,42 €**.

Recurso	**Coste prorrateado (€)**
Alquiler	4.765,89
Electricidad	8,40
Internet	23,42
Total (€)	**4.797,71**

Tabla 2.4: Tabla de costes infraestructurales.

2.5. Control de riesgos

La naturaleza de las auditorías de carácter general suelen tener un procedimiento: en un plazo máximo establecido por el cliente (generalmente dos semanas), se tiene que auditar todo lo posible el dispositivo ofrecido. Debido a ello y a que varias subtareas no se pudieron realizar, ya sea porque fallaba o faltaba el material, la planificación inicial ha sufrido ciertos cambios. Como **medida de prevención** de estas desviaciones, se planeó el plan de contingencia mostrado en la Tabla 2.5.

La **prioridad** del plan de contingencia es ceñirse al máximo posible a la **planificación inicial** del proyecto y evitar desviaciones drásticas. También se procura la **salud** del alumno y del supervisor en caso que esta se pueda ver comprometida.

Categoría	Código del riesgo	Descripción	Solución
Material	RM01.1	Material no disponible	Reorganizar subtareas a la espera de que esté disponible
	RM01.2	Material no disponible	Descartarlo y buscar uno nuevo
	RM02	Material no compatible	Intercambiarlo por uno nuevo que sea compatible
	RM03	Material con problemas	Intercambiarlo por uno nuevo que no tenga problemas
Salud	RS01.1	Uso del soldador	Procurar no manipular la punta mientras esté encendido o caliente
	RS01.2	Uso del soldador	Utilizar un área ventilada para soldar
	RS01.3	Uso del soldador	Utilizar una alfombrilla para proteger la superficie donde se está soldando
	RS01.4	Uso del soldador	Utilizar extractor de humos tóxicos del estaño
Tiempo	RT01.1	Falta de tiempo	Reorganizar tareas para tener la posibilidad de terminarlo en más tiempo
	RT01.2	Falta de tiempo	Dejarlo de lado, respetando la naturalidad de las auditorías
	RT02.1	Sobra de tiempo	Reorganizar tareas para adelantar el trabajo
	RT02.2	Sobra de tiempo	Revisar y adelantar la documentación
Conocimiento	RCE01.1	Falta de conocimiento	Investigar en Internet acerca de cualquier fuente que alimente el conocimiento necesario
	RCE01.2	Falta de conocimiento	Preguntar al supervisor cualquier duda necesaria
	RCE02.1	Falta de experiencia	Investigar en Internet acerca de cualquier fuente que registre y documente previa experiencia de otros usuarios
	RCE02.2	Falta de experiencia	Solicitar ayuda al supervisor

Tabla 2.5: Tabla de posibles riesgos para el proyecto.

2.6. Planificación temporal

La planificación inicial consta de cuatro bloques de semanas:

- Primera semana: **toma de contacto**. Se empleará para adaptarse a la empresa, planificar más detalladamente el proyecto formativo, comentar todas las tecnologías a utilizar durante la estancia y empezar una formación base teórica de esas tecnologías.

- Siguientes tres semanas: **formación**. Esta formación se realizaría utilizando enrutadores. Se emplearán para identificar los componentes de un dispositivo hardware y averiguar la codificación de los puertos de conexión que posee, además de aprender el proceso de volcado de memoria. También se investigará acerca del firmware del enrutador y a como desensamblarlo mediante GHidra. La parte final de esta formación constaría de realizar una auditoría a la web de configuración del enrutador.

- Siguientes cuatro semanas: **cerradura inteligente**. Se dedicaría a aplicar todos los conocimientos aprendidos a una nueva cerradura. Se comenzará con la identificación de puertos y del hardware que posee seguido de verificar la seguridad del firmware de la cerradura (en caso que lo tenga). Después se investigará acerca de las posibles vulnerabilidades de los protocolos inalámbricos NFC (del inglés *Near Field Communication*, Comunicación de en Áreas Muy Cercanas) y BLE (del inglés *Bluetooth Low Energy*, comunicación Bluetooth de Baja Energía) y del protocolo de huella dactilar. Finalmente, se tratará de auditar la aplicación móvil de la cerradura.

- Últimas cuatro semanas: **proyecto final, PLC (del inglés *Power Line Communications*, Comunicaciones vía Red Eléctrica)**. Estas últimas semanas pondrían a prueba todos los conocimientos aprendidos en las semanas de formación y las semanas empleadas en la cerradura inteligente en un nuevo dispositivo de red PLC.

La Tabla 2.6 detalla más concretamente las tareas, con una breve descripción de cada una de las subtareas que conlleva cada tarea, y las fechas y cantidad de días que se planean tener en cuenta para realizar cada una de las subtareas. Cada bloque de semanas tiene su día final dedicado a la documentación de todo lo realizado. Los días festivos han sido tenidos en cuenta para las duraciones de las semanas y las tareas. La Figura 2.1 muestra un **diagrama de Gantt** que resume toda la planificación del proyecto.

Tarea(s)	Descripción subtarea(s)	Duración
1. Toma de contacto con la empresa Proceso de onboarding realizado en la empresa	**1.1 Formación inicial** El supervisor ofrecerá enlaces de documentación para ser formado desde el inicio de la estancia en prácticas	27/09 - 01/10 (una semana)
2. Formación previa Formación base acerca de todo lo relacionado con el firmware y posibles protocolos a utilizar	**2.1 Investigación firmware** A partir de un firmware de un enrutador específico intentar investigar con las herramientas necesarias el firmware para obtener su información	04/10-06/10 (tres días)
	2.2 Desensamblado de firmware Uso del programa GHidra para desensamblar el firmware y poder detectar posibles bloques de código donde se puedan encontrar fallos	07-10-08/10 (dos días)
	2.3 Formación auditoría página web Realizar una auditoría a la página web de configuración del enrutador y encontrar posibles fallos	11/10-14/10 (tres días)
	2.4 Identificación de componentes Apertura de un router para investigar su PCB e identificar los componentes que posee: Chips, memorias, test points, etc.	15/10-16/10 (dos días)
	2.5 Investigación de comunicaciones físicas Investigación de los protocolos de conexión mediante los puertos Serie, UART y SPI	19/10-22/10 (cuatro días)
	2.6 Documentación Documentar todo lo trabajado	23/10 (un día)
3. Auditorías Se realizarán auditorías a una cerradura inteligente para investigar todas sus vulnerabilidades en los protocolos que posea. Se procurará que la cerradura disponga de todos los protocolos que se mencionan en las subtareas	**3.1 Auditar Firmware** Auditar la seguridad del firmware de la cerradura e investigar sus puertos de comunicación hardware	25/10-27/10 (tres días)
	3.2 Auditar NFC Auditoría del protocolo NFC de la cerradura	28/10-03/11 (cuatro días)
	3.3 Auditar BLE Auditoría del protocolo BLE de la cerradura.	04/11-10/11 (cinco días)
	3.4 Auditar huella dactilar Auditoría del protocolo de autenticación mediante la huella dactilar de la cerradura.	11-11/15/11 (tres días)
	3.5 Auditar aplicación móvil y protocolos WiFi Auditoría a la aplicación móvil de la cerradura. También se investigará acerca de sus protocolos de comunicación vía WiFi	16/11-19/11 (cuatro días)
	3.6 Documentación Documentar todo lo trabajado	20/11 (un día)
4. Proyecto final Aplicar todo lo aprendido en un nuevo dispositivo de red IoT	**4.1 Auditoría de PLC** Se realizará un proceso de auditoría por cuenta propia a un dispositivo de red PLC	22/11-21/12 (cuatro semanas)

Tabla 2.6: Tabla de tareas y subtareas.

2.7. Seguimiento del proyecto

Durante la estancia en prácticas se ha mantenido **contacto periódico** tanto con el supervisor como con el profesor.

Con el supervisor de la empresa, Manuel Ginés Rodríguez, se realizaban **comunicaciones**

y reuniones telemáticas cada día o cada vez que fuera necesario vía Microsoft Teams, la aplicación utilizada por la empresa para comunicarse internamente. También se mantenían **reuniones presenciales** cada día martes para preparar las tareas a realizar durante cada semana y otorgar el material necesario.

Mientras que con el profesor tutor, Manuel Francisco Dolz Zaragozá, se realizaban **reuniones cada dos semanas** para explicar y comentar todo aquello realizado durante cada quincena. Se le avisaba de todo el trabajo realizado, las desviaciones que hubieran sucedido, el trabajo futuro y cualquier duda que el alumno tuviera respecto a la realización de la estancia en prácticas o de la redacción de las quincenas.

Naturalmente ocurrieron **desviaciones** en relación a la planificación del proyecto debido a que es un proyecto con metodología predictiva. Estas desviaciones se redactaron y explicaron a medida que el alumno realizaba sus entregas quincenales de avance en el proyecto.

Para comenzar, las tres semanas de **formación** se extendieron a **cuatro semanas** en total. Al inicio de estas semanas de formación se esperaba comenzar con el estudio de los puertos serie y las comunicaciones Bluetooth, pero debido a **problemas disponibilidad** del material y de su **compatibilidad** se reorganizó de manera que se comenzara con el desensamblado del firmware y dejar de lado la puesta en práctica de la formación BLE. Se procedió a realizar la auditoría con el enrutador y la cerradura inteligente especificados dentro del bloque de semanas de formación, utilizando así estos dispositivos para poner en práctica todo lo aprendido al inicio de este bloque de semanas en lugar de reservar un bloque de tiempo específico para cada uno de ellos.

Además, la adición de esa semana permitiría el **aprendizaje de auditorías de aplicaciones móviles**. Esta semana extra serviría de gran ayuda debido a que la empresa Sofistic contactó con la **organización ioXt Alliance** para poder auditar una aplicación móvil orientada a dispositivos IoT de una de las empresas que colabora con esta organización. Esta auditoría se llevaría a cabo en un periodo de dos semanas, dónde además su comienzo coincide con la finalización del periodo de formación.

Respecto al periodo de **auditoría de la aplicación**, como se hizo en conjunto con el supervisor, se realizó en menos tiempo del estimado, en **poco más de una semana**. Para no quedar a la espera del comienzo del periodo final, el de la auditoría del dispositivo PLC, se decidió reorganizar ese periodo de manera que se comenzaría con la introducción hacia la auditoría de este dispositivo. Por tanto, este periodo se extendería unos días más de las cinco semanas previstas para este periodo.

De esta manera, el **periodo final** sería dedicado a auditar el PLC por el tiempo mencionado anteriormente. Con ello, finalmente, se completa todo el tiempo de estancia en prácticas, haciendo de este dispositivo el **proyecto final**.

La Figura 2.2 muestra un nuevo **diagrama de Gantt** que recopila la **nueva planificación**, respetando las desviaciones ocurridas.

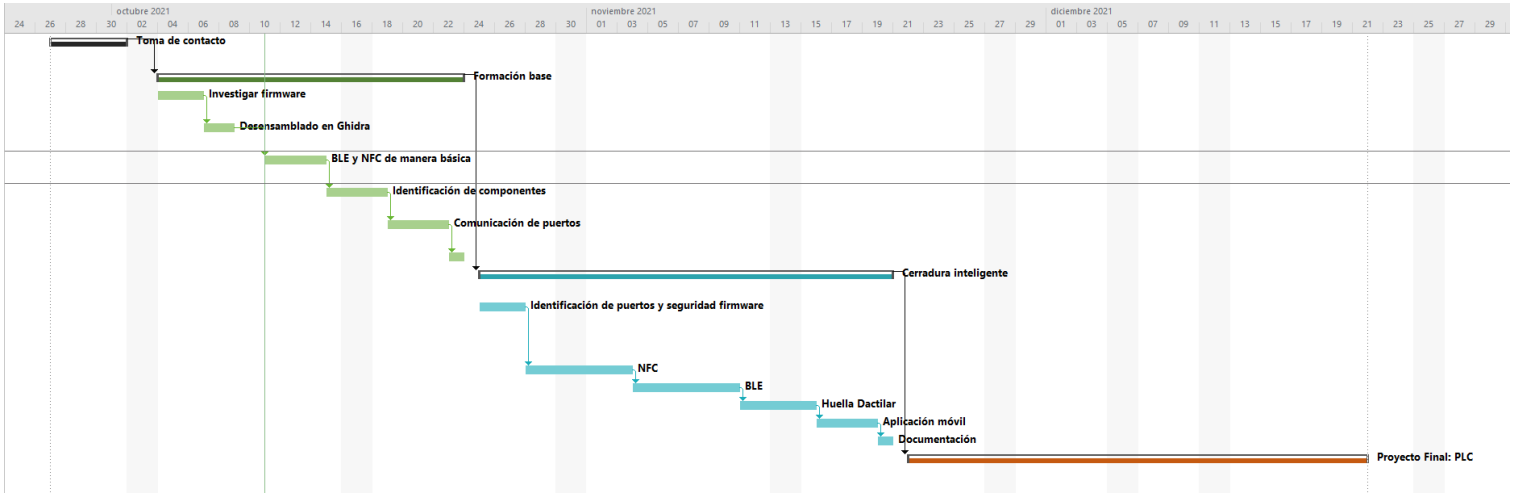

Figura 2.1: Diagrama de Gantt del proyecto formativo.

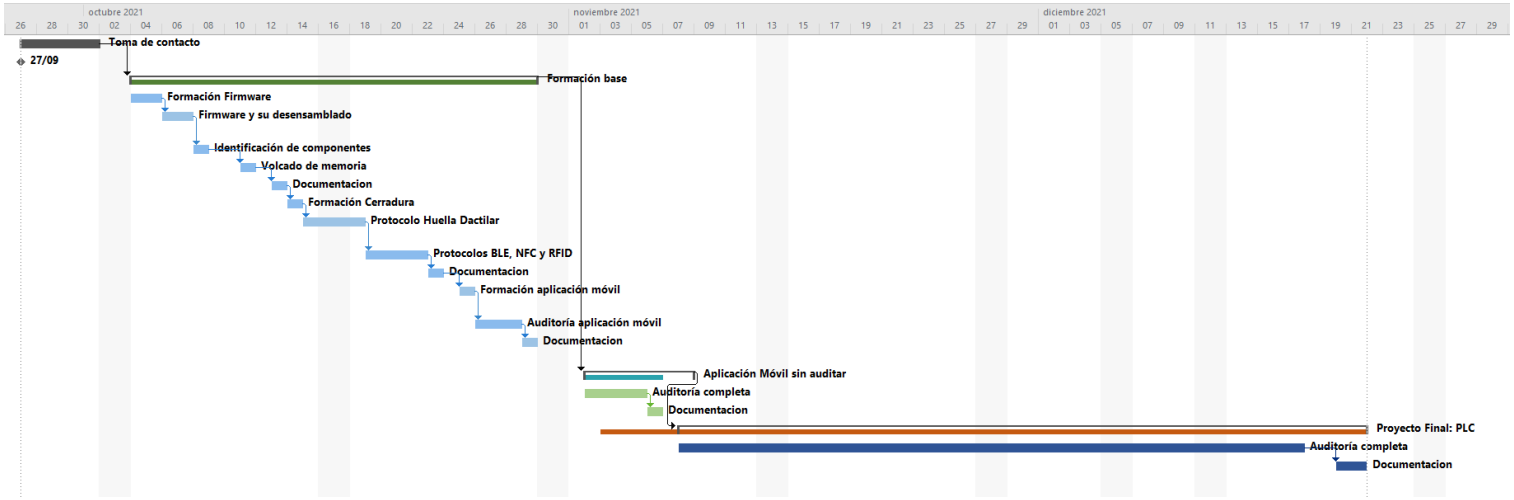

Figura 2.2: Diagrama de Gantt actualizado del proyecto formativo.

Capítulo 3

Análisis y diseño del procedimiento de la auditoría

En este capítulo se definirán los **requisitos** del proceso de auditoría, junto con un pequeño análisis de estos. También se realizará el **diseño** del proceso de auditoría mediante casos de uso junto con una descripción que la explique.

3.1. Identificación de requisitos

Para realizar un correcto **análisis** del procedimiento de la auditoría, es necesario **definir los requisitos**. Los requisitos son las **necesidades** que debe ofrecer y cumplir la auditoría de un dispositivo IoT. Al ser un diseño y desarrollo de una metodología y no de un sistema software, los requisitos de este proyecto se catalogan en: **requisitos de usuario** y **requisitos de la metodología de auditoría**

3.1.1. Requisitos de usuario

Los **requisitos de usuario** en base a una metodología de auditoría se basan en todos los pasos que el usuario final, en este caso el auditor de una empresa, debe realizar para llevar a cabo una auditoría de manera correcta. Las siguientes 7 tablas (desde la Tabla 3.1 hasta la Tabla 3.7 incluidas) recopilan estos requisitos.

Código	REQ-U-1
Nombre	Identificar el dispositivo
Descripción	La persona auditora debe aplicar todos los requisitos relacionados con la identificación del dispositivo
Prioridad	Alta

Tabla 3.1: Primer requisito de usuario.

Código	REQ-U-2
Nombre	Definir el vector de ataque
Descripción	La persona auditora debe aplicar todos los pasos relacionados con la definición del vector de ataque
Prioridad	Alta

Tabla 3.2: Segundo requisito de usuario.

Código	REQ-U-3
Nombre	Realizar la auditoría remota
Descripción	La persona auditora debe llevar a cabo la auditoría remota siguiendo todos los requisitos necesarios para ello
Prioridad	Alta

Tabla 3.3: Tercer requisito de usuario.

Código	REQ-U-4
Nombre	Realizar la auditoría local
Descripción	La persona auditora debe ser capaz de llevar a cabo la auditoría local aplicando todos los pasos necesarios
Prioridad	Alta

Tabla 3.4: Cuarto requisito de usuario.

Código	REQ-U-5
Nombre	Realizar la auditoría física
Descripción	La persona auditora debe llevar a cabo la auditoría física aplicando todos los pasos necesarios para ello
Prioridad	Alta

Tabla 3.5: Quinto requisito de usuario.

Código	REQ-U-6
Nombre	Resumir y puntuar vulnerabilidades
Descripción	La persona auditora debe ser capaz de redactar un resumen de todas las vulnerabilidades encontradas y puntuar su gravedad
Prioridad	Alta

Tabla 3.6: Sexto requisito de usuario.

Código	REQ-U-7
Nombre	Redactar informe final
Descripción	La persona auditora debe llevar a cabo todos los pasos necesarios para redactar un resumen ejecutivo
Prioridad	Alta

Tabla 3.7: Séptimo requisito de usuario.

3.1.2. Requisitos de la metodología de auditoría

Los **requisitos de la metodología de auditoría** se establecen según los **pasos y objetivos que debe cumplir** este proceso para poder realizar una auditoría sobre un dispositivo IoT. Se debe realizar un proceso genérico, debido a que no todos los dispositivos IoT disponen de los mismos protocolos o funciones. Por tanto, se trata de realizar una metodología que englobe la **máxima variedad de estos dispositivos**. Los requisitos para el proceso que se desarrolla en este proyecto se presentan en las siguientes 21 tablas (desde la Tabla 3.8 hasta la Tabla 3.28 incluidas) que contienen el **código** del requisito, un **nombre** explicativo del requisito en cuestión, una breve **descripción** y la **prioridad** de este. También se han clasificado en grupos según el proceso al que pertenece cada requisito.

Código	REQ-MA-1-1
Nombre	Identificar funcionalidades del dispositivo
Descripción	Se identifican las funcionalidades que cumplen con el propósito del dispositivo
Prioridad	Alta

Tabla 3.8: Identificar dispositivo: Primer requisito.

Código	REQ-MA-1-2
Nombre	Identificar protocolos del dispositivo
Descripción	Se identifican los medios tecnológicos mediante los cuales consigue su propósito
Prioridad	Alta

Tabla 3.9: Identificar dispositivo: Segundo requisito.

Código	REQ-MA-1-3
Nombre	Identificar aplicaciones web relacionadas con el dispositivo
Descripción	Se investiga acerca del dispositivo por si dispone de alguna aplicación propia, tanto web o móvil
Prioridad	Alta

Tabla 3.10: Identificar dispositivo: Tercer requisito.

Código	REQ-MA-1-4
Nombre	Identificar funcionalidades extra
Descripción	Se identifican otras funciones que pueda disponer, adicionales a las que cumplen con el propósito del dispositivo
Prioridad	Alta

Tabla 3.11: Identificar dispositivo: Cuarto requisito.

Código	REQ-MA-2-1
Nombre	Definir un vector de ataque remoto
Descripción	Se definen todos los ataques posibles que se puedan realizar al dispositivo desde el punto de vista remoto, es decir, como si el dispositivo no estuviera disponible o no se tuviera acceso a él
Prioridad	Alta

Tabla 3.12: Definir vectores de ataque: Primer requisito.

Código	REQ-MA-2-2
Nombre	Definir un vector de ataque local
Descripción	Se definen todos los ataques posibles que se puedan realizar al dispositivo desde el punto de vista local, es decir, una vez el dispositivo esté disponible y se tenga acceso a él
Prioridad	Alta

Tabla 3.13: Definir vectores de ataque: Segundo requisito.

Código	REQ-MA-2-3
Nombre	Definir un vector de ataque físico
Descripción	Se definen todos los ataques posibles que se puedan realizar al dispositivo desde el punto de vista físico, es decir, manipulando físicamente el dispositivo (por ejemplo, desensamblarlo)
Prioridad	Alta

Tabla 3.14: Definir vectores de ataque: Tercer requisito.

Código	REQ-MA-3-1
Nombre	Investigar y auditar firmware del dispositivo
Descripción	Se investiga si el firmware del dispositivo se encuentra disponible públicamente y, en caso afirmativo, se auditan
Prioridad	Alta

Tabla 3.15: Auditoría remota: Primer requisito.

Código	REQ-MA-3-2
Nombre	Investigar y auditar aplicaciones web del dispositivo
Descripción	Se investiga si existen aplicaciones, móviles o web, que estén relacionadas con el dispositivo y, en caso afirmativo, se auditan
Prioridad	Alta

Tabla 3.16: Auditoría remota: Segundo requisito.

Código	REQ-MA-3-3
Nombre	Investigar y auditar protocolos de red externos del dispositivo
Descripción	Se investiga si el dispositivo dispone de una conexión hacia el exterior o de una red generada por él mismo y, en caso afirmativo, se auditan todos los protocolos y comunicaciones que disponga
Prioridad	Alta

Tabla 3.17: Auditoría remota: Tercer requisito.

Código	REQ-MA-4-1
Nombre	Investigar y auditar protocolos de red internos del dispositivo
Descripción	Se investiga si el dispositivo dispone de conexiones de red internas, sean o no inalámbricas, y, en caso afirmativo, se auditan todos los protocolos y comunicaciones que disponga
Prioridad	Alta

Tabla 3.18: Auditoría local: Primer requisito.

Código	REQ-MA-4-2
Nombre	Investigar y auditar el protocolo NFC
Descripción	Investigar si el dispositivo dispone de protocolos y tarjetas o llaves NFC y, en caso afirmativo, se auditan
Prioridad	Alta

Tabla 3.19: Auditoría local: Segundo requisito.

Código	REQ-MA-4-3
Nombre	Investigar y auditar el protocolo Bluetooth y BLE
Descripción	Investigar si el dispositivo dispone de protocolos Bluetooth o Bluetooth Low Energy (BLE) y, en caso afirmativo, se auditan
Prioridad	Alta

Tabla 3.20: Auditoría local: Tercer requisito.

Código	REQ-MA-4-4
Nombre	Investigar y auditar el protocolo RFID y radiofrecuencias
Descripción	Investigar si el dispositivo dispone de protocolos mediante radiofrecuencias y/o RFID y, en caso afirmativo, se auditan
Prioridad	Alta

Tabla 3.21: Auditoría local: Cuarto requisito.

Código	REQ-MA-5-1
Nombre	Investigar la placa base del dispositivo e identificar componentes
Descripción	Se investigan y analizan los componentes del dispositivo, como el microprocesador, las memorias, testpoints o algún otro componente relacionado con el propósito del dispositivo
Prioridad	Alta

Tabla 3.22: Auditoría física: Primer requisito.

Código	REQ-MA-5-2
Nombre	Investigar y auditar interfaz serie, debug y testpoints del dispositivo
Descripción	Se investiga si la placa base del dispositivo dispone de una serie de pines o puertos en serie (que suelen ser interfaces de comunicación serie, de debug o testpoints) y, en caso afirmativo, se auditan
Prioridad	Alta

Tabla 3.23: Auditoría física: Segundo requisito.

Código	REQ-MA-5-3
Nombre	Investigar y auditar el firmware instalado y el contenido de las memorias
Descripción	En caso de no encontrar el firmware de manera pública, se puede encontrar en las memorias del dispositivo, por tanto se procede a auditar las memorias
Prioridad	Alta

Tabla 3.24: Auditoría física: Tercer requisito.

Código	REQ-MA-5-4
Nombre	Investigar y auditar los sensores instalados en el dispositivo y sus protocolos o medios de comunicación con el microprocesador
Descripción	Se investigan si existen algunos sensores que ayuden al cumplimiento del propósito del dispositivo y, en caso afirmativo, se auditan
Prioridad	Alta

Tabla 3.25: Auditoría física: Cuarto requisito.

Código	REQ-MA-5-5
Nombre	Investigar y auditar el sistema de arranque y el sistema operativo del dispositivo
Descripción	Se investigan y se auditan el sistema de arranque y el sistema operativo del dispositivo para obtener control del dispositivo
Prioridad	Alta

Tabla 3.26: Auditoría física: Quinto requisito.

Código	REQ-MA-6-1
Nombre	Resumir y puntuar la gravedad de las vulnerabilidades encontradas
Descripción	Se resumen todas las vulnerabilidades encontradas en cada paso de la auditoría y se puntúa la gravedad de estas en base a unos criterios de clasificación oficiales
Prioridad	Alta

Tabla 3.27: Puntuar vulnerabilidades: Primer requisito.

Código	REQ-MA-7-1
Nombre	Redactar un informe final
Descripción	Se redacta un resumen ejecutivo para la empresa del dispositivo auditado que incluya todas las vulnerabilidades encontradas, su gravedad y cómo afecta al usuario poseedor del dispositivo, junto con todo ello que no dio tiempo a auditar
Prioridad	Alta

Tabla 3.28: Redactar informe final: Primer requisito.

3.2. Diseño del proceso de auditoría

Para realizar el **diseño** del proceso de auditoría, se utilizan los **diagramas de casos de uso**. Estos diagramas se utilizan para representar los pasos que se tienen que realizar en un proceso de auditoría. Además, se utilizan **diagramas de flujo** para dar un ejemplo al uso de los diagramas de casos de uso establecidos.

3.2.1. Diagramas de casos de uso

Los **diagramas de casos de uso** se representan mediante **diagramas UML** (del inglés *Unified Modeling Language*, Lenguaje de Modelado Unificado) para una mayor facilidad a la hora de entender los pasos que se abordan en el proceso de auditoría. En estos diagramas existen **actores**, personajes o entidades que realizan **actividades** o acciones, enlazadas con ellos. Para este proyecto, las **actividades** son todos los pequeños pasos que se tienen que realizar para llevar a cabo una auditoría, los cuales **vienen derivados de los requisitos** del proyecto ya establecidos. El **actor** es la **persona auditora** que aplicará cada una de las actividades para completar el proceso de auditoría.

El primer caso es el de **identificar el dispositivo**. Para ello, la persona auditora debe cumplir los requisitos **REQ-MA-1-1**, **REQ-MA-1-2**, **REQ-MA-1-3** y **REQ-MA-1-4**. En la Figura 3.1 se muestra el diagrama para este caso de uso.

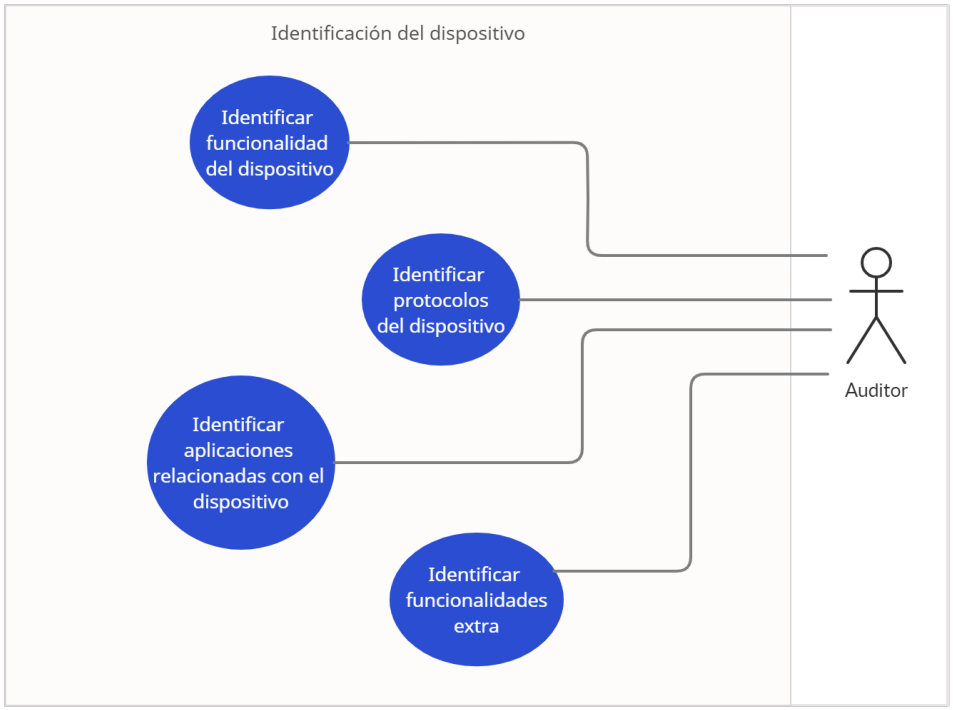

Figura 3.1: **CU01**: Diagrama de caso de uso para **identificar el dispositivo**.

El segundo caso es el de **definir el vector de ataque**. La persona auditora debe cumplir los requisitos **REQ-MA-2-1**, **REQ-MA-2-2** y **REQ-MA-2-3**. En la Figura 3.2 se muestra el diagrama para este caso de uso.

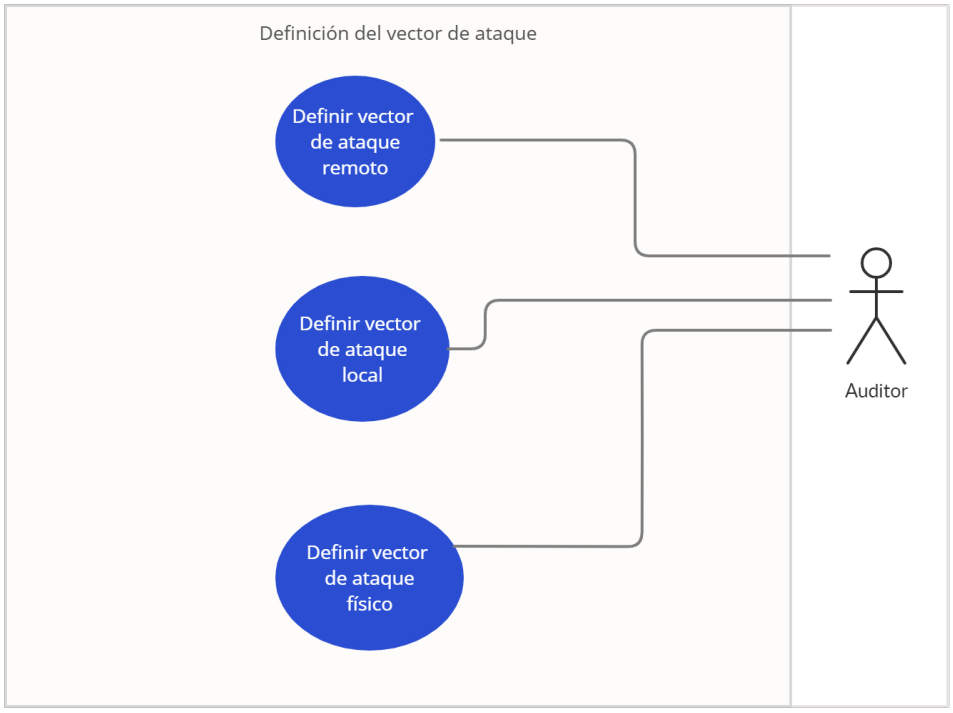

Figura 3.2: **CU02**: Diagrama de caso de uso para **definir el vector de ataque**.

Respecto al tercer caso, se trata de **realizar una auditoría remota**. Para ello, la persona auditora debe completar los requisitos **REQ-MA-3-1**, **REQ-MA-3-2** y **REQ-MA-3-3**. En la Figura 3.3 se muestra el diagrama para este caso de uso.

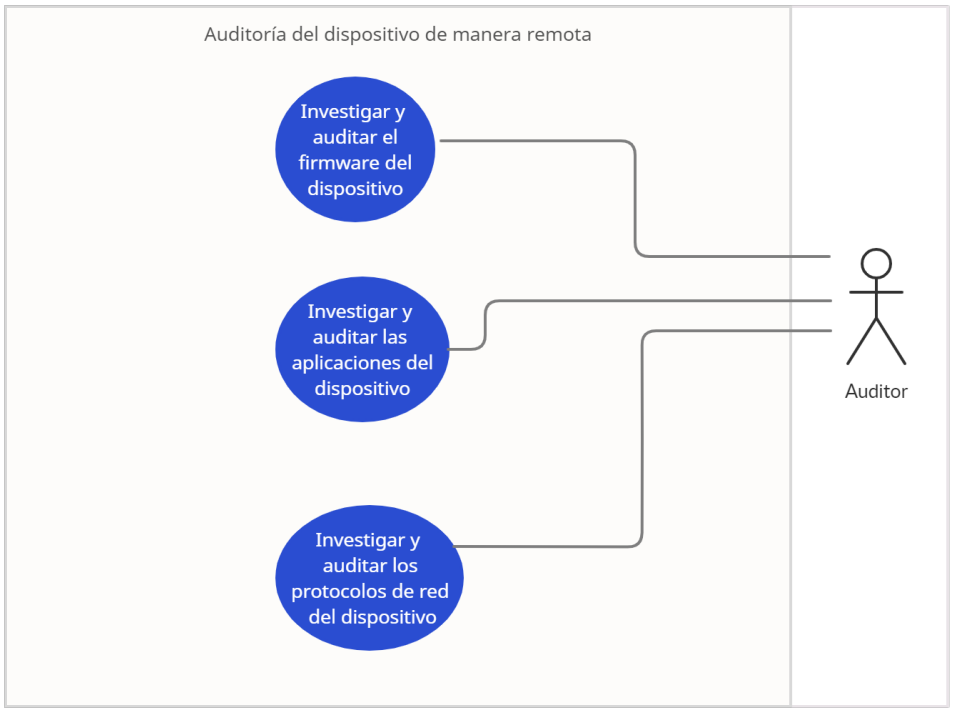

Figura 3.3: **CU03**: Diagrama de caso de uso para **realizar una auditoría remota**.

En cuanto al cuarto caso, hay que **realizar una auditoría local**. La persona auditora debe cumplir los requisitos **REQ-MA-4-1**, **REQ-MA-4-2**, **REQ-MA-4-3** y **REQ-MA-4-4**. El diagrama de este caso de uso se muestra en Figura 3.4.

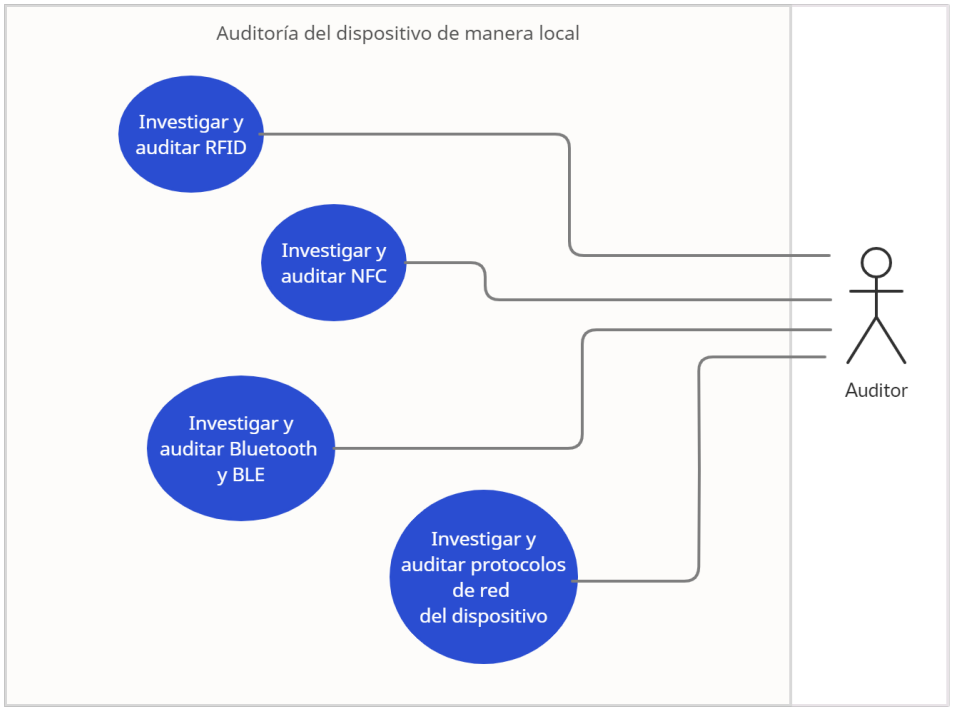

Figura 3.4: **CU04**: Diagrama de caso de uso para **realizar una auditoría local**.

El siguiente caso, el quinto, es el de **realizar una auditoría física**. Los requisitos **REQ-MA-5-1**, **REQ-MA-5-2**, **REQ-MA-5-3**, **REQ-MA-5-4** y **REQ-MA-5-5** se deben cumplir en este caso de uso. En la Figura 3.5 se muestra su diagrama.

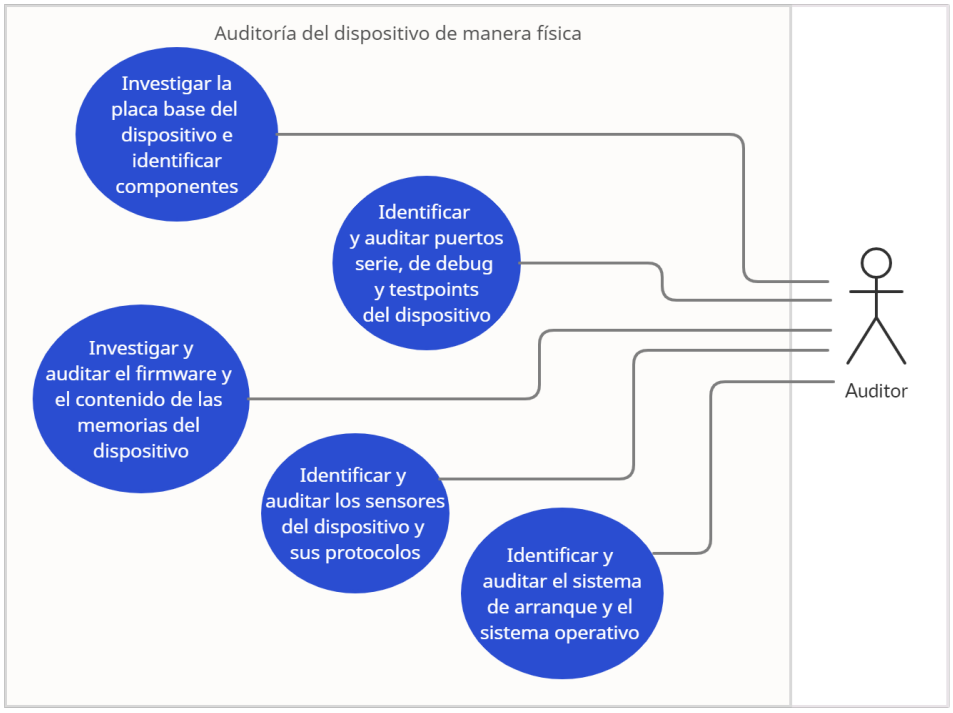

Figura 3.5: **CU05**: Diagrama de caso de uso para **realizar una auditoría física**.

Para el sexto caso, el de **resumir y puntuar las vulnerabilidades**, se tiene que cumplir el requisito **REQ-MA-6-1**. En la Figura 3.6 se muestra el diagrama para este caso de uso.

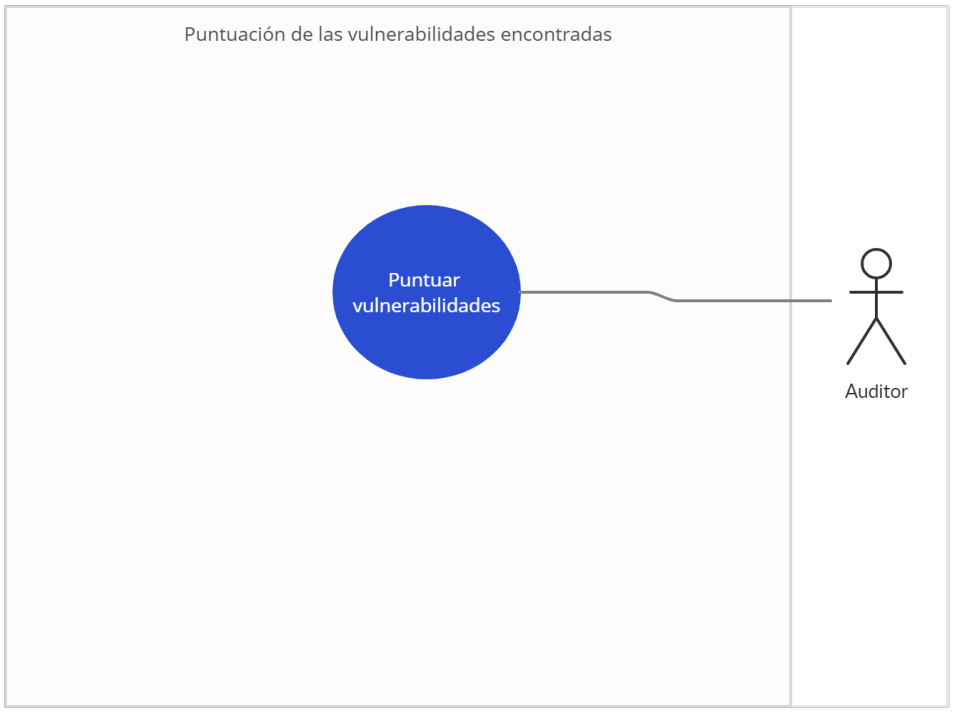

Figura 3.6: **CU06**: Diagrama de caso de uso para **resumir y puntuar las vulnerabilidades**.

Finalmente, para el séptimo caso, se trata de **redactar un informe**. Para ello, la persona auditora debe cumplir el requisito **REQ-MA-7-1**, dónde la Figura 3.7 muestra su diagrama.

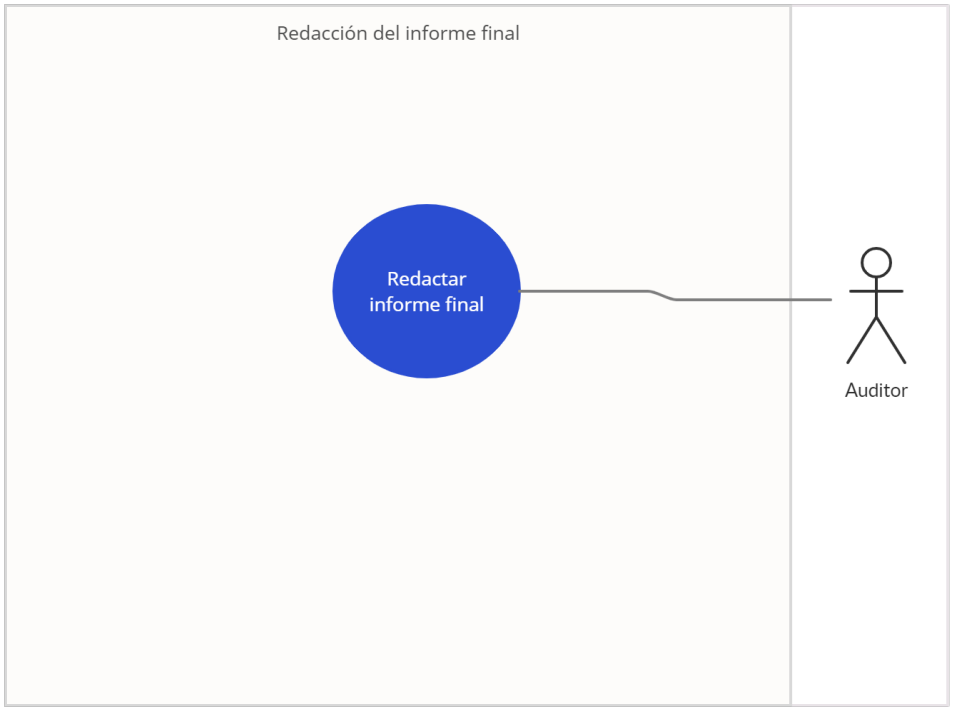

Figura 3.7: **CU07**: Diagrama de caso de uso para **redactar el informe final**.

3.2.2. Diagramas de flujo

Para representar de una manera más sencilla la metodología de auditoría diseñada, se implementa el diagrama de flujo de la Figura 3.8. En esta se describe desde el inicio como se debe abordar el comienzo de la auditoría, seguido de qué pasos tomar según los resultados encontrados, siempre llegando a la conclusión de redactar un informe final.

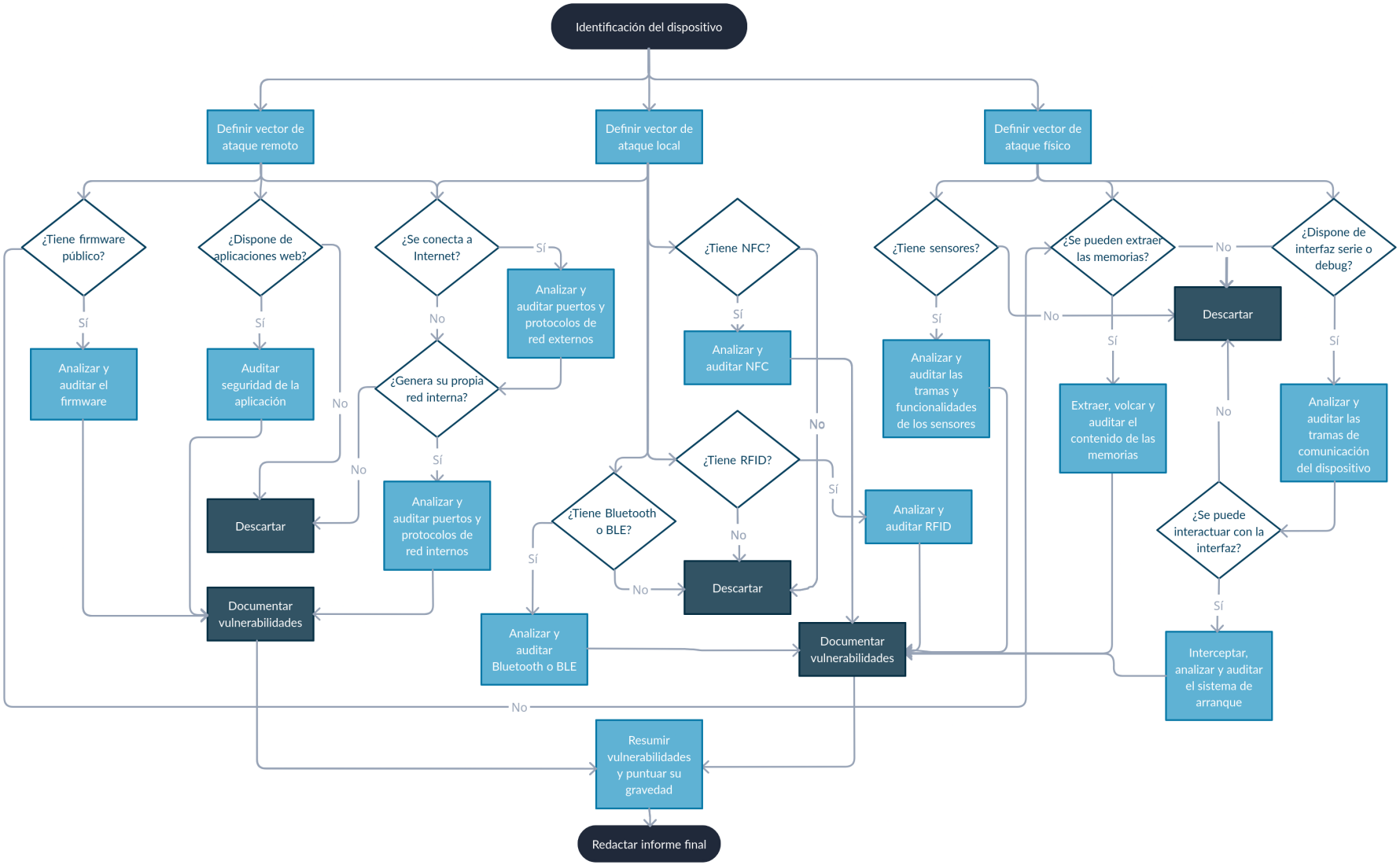

Figura 3.8: Diagrama de flujo de la metodología de auditoría.

3.3. Definición y descripción de la metodología de auditoría

La **metodología de auditoría** desarrollada se basa en los **requisitos establecidos**. Cabe destacar que la naturaleza de una auditoría de seguridad se debe realizar en un plazo establecido con el cliente que solicita la auditoría. Este plazo generalmente es de dos semanas, y se procura auditar al máximo lo solicitado por el cliente. Mientras se esté realizando la auditoría, es importante **documentar** todos los procesos y pasos que se estén realizando para cada requisito. Esto resultará de gran ayuda para el último requisito, que será la redacción del informe final con todas las vulnerabilidades resumidas y puntuadas para el cliente.

3.3.1. Identificación del dispositivo

Para comenzar esta metodología, primero se debe **identificar el dispositivo** que se está auditando y su propósito. En primer lugar, se identifican todas las **funcionalidades del dispositivo** que ayuden a cumplir con **su propósito**. Por ejemplo, si dispone de algún panel táctil o algún sensor que se utilice para cumplir su finalidad. Esto representa el requisito **REQ-MA-1-1**.

A continuación se deben identificar los **protocolos** que utiliza para comunicarse con el usuario del dispositivo. Por ejemplo, si se trata de una comunicación vía WiFi, Bluetooth o NFC. Esto representa el requisito **REQ-MA-1-2**. También es necesario investigar si tiene alguna **aplicación externa** que ayude a la configuración, gestión o uso del dispositivo, sea aplicación móvil o web, para cumplir con el requisito **REQ-MA-1-3**.

Finalmente, se puede investigar si el dispositivo dispone de **funcionalidades adicionales** que faciliten el cumplimiento del propósito de la aplicación, o que simplemente sean extras añadidos al dispositivo. Esto viene definido en el requisito **REQ-MA-1-4**, aunque puede no obtener resultados.

Todo lo identificado en esta primera parte de la metodología debe quedar **documentado**, ya que se necesitará para proseguir con el siguiente apartado.

3.3.2. Definición del vector de ataque

El **vector de ataque** recopila todos los puntos desde el cual un dispositivo puede ser auditado. En el caso de los dispositivos IoT, se divide este vector en **tres vertientes**: vector de ataque **remoto**, **local** y **físico**.

- El **vector de ataque remoto**, que representa el requisito **REQ-MA-2-1**, contiene todos los puntos de ataque que se puedan realizar **sin tener acceso al dispositivo**, es decir, sin estar conectado de ninguna manera a este. Algunos puntos donde se pueden encontrar vulnerabilidades son: el firmware del dispositivo, aplicaciones relacionadas con el dispositivo o los protocolos de red del dispositivo.
- El **vector de ataque local**, que representa el requisito **REQ-MA-2-2**, engloba todos los puntos de ataque que se puedan realizar de manera local, es decir, con acceso al dispositivo

para tener **conexión directa** con él. Pero no se debe manipular en exceso, solamente para **utilizarlo** de manera que cumpla con **su propósito**.

- El **vector de ataque físico**, que representa el requisito **REQ-MA-2-3**, define todos los puntos de ataque que se puedan ejecutar **manipulando físicamente** el dispositivo. Esto es, desensamblar el dispositivo para acceder a todos sus componentes hardware internos. De esta manera, se pueden identificar su microprocesador, sus memorias o los sensores que disponga el dispositivo.

Una vez identificados todos los **puntos de ataque** de cada vector, se debe comenzar con el ataque remoto. Seguido de ello, el ataque local y, finalmente, el físico. Esto se debe a que, **en** este **orden**, se minimiza el riesgo de estropear el dispositivo durante la auditoría y garantiza el cumplimiento del proceso de auditoría sin riesgos.

3.3.3. Auditoría remota

Desde el punto de **ataque remoto** se encuentran diversos objetivos.

- En primer lugar, se puede investigar por Internet si existe algún **firmware** del dispositivo (**REQ-MA-3-1**). Generalmente suelen ser públicos por parte de la marca propietaria. En caso de no disponer del firmware público, no se realiza este paso. Por contrario, si se dispone de el, se deberá analizar adecuadamente para encontrar todo tipo de posibles vulnerabilidades o analizar cómo dicta el funcionamiento del dispositivo.
- Además del firmware, también se pueden investigar las posibles **aplicaciones web** que pueda utilizar (**REQ-MA-3-2**). De manera general, suelen utilizar aplicaciones móviles para ayudar al usuario a gestionar y configurar de manera más fácil el dispositivo. Pero estas aplicaciones también pueden contener vulnerabilidades, por tanto también se deben auditar en caso que estén disponibles.
- Finalmente, en caso de ser un dispositivo que pueda comunicarse con el exterior, es decir con algún servidor en Internet gestionado por la marca del dispositivo, se deben analizar los **protocolos** y las **comunicaciones** que realiza para comprobar que sean seguras (**REQ-MA-3-3**). Además, se pueden investigar los **puertos de red** [54] que estén disponibles y analizar si conllevan riesgos [3] a la red o no.

3.3.4. Auditoría local

Respecto al **ataque local**, se dispone de más variedad de puntos donde encontrar vulnerabilidades.

- Desde este ataque se puede proseguir con el ataque a **protocolos de red** del ataque remoto (**REQ-MA-4-1**). De esta manera, se analizan todas las conexiones y comunicaciones realizadas dentro de la **red interna**, sean o no de manera inalámbrica. También se pueden auditar todas las comunicaciones internas realizadas a través de los **puertos de red** que se han analizado, sea en este ataque o en el anterior.

- Además, debido al avance de las nuevas tecnologías, se pueden encontrar dispositivos con el sistema **NFC** [53] (del inglés *Near-Field Communication*, Comunicación en Áreas Muy Cercanas) incorporado (**REQ-MA-4-2**). Se trata de una tecnología inalámbrica de corto alcance que permite transmitir e interpretar información entre dispositivos de manera rápida y sencilla. Puede estar presente en tarjetas de acceso, de crédito o en móviles. Por tanto, es una tecnología donde sus vulnerabilidades pueden ser críticas.

- Otra tecnología inalámbrica que pueden incorporar es **Bluetooth**, o su versión de menor consumo **BLE** (**REQ-MA-4-3**). **BLE** [4] (del inglés *Bluetooth Low Energy*, comunicación Bluetooth de Baja Energía) es un medio de comunicación radiomagnético de baja potencia utilizado para transmitir datos hasta en 40 canales dentro de las bandas de 2.4 GHz (Gigahercios) y del rango de frecuencias permitidas en Europa. Esta tecnología permite flexibilidad a la hora de comunicar dispositivos, sea entre dos de ellos o varios. Para encontrar vulnerabilidades en este sistema, se pueden interceptar sus comunicaciones y analizar las tramas enviadas entre los dispositivos. De esta manera se consigue averiguar si la comunicación entre los dispositivos se establece y se realiza de manera segura y solamente entre ellos dos.

- Finalmente, otro medio inalámbrico que pueden incorporar estos dispositivos es el **RFID** [55] (del inglés *Radio Frequency Identification*, Identificación por Radio Frecuencia) (**REQ-MA-4-4**). Este medio establece sus comunicaciones en base al emparejamiento electromagnético o electrostático, incluido en un rango de frecuencias definido dentro del espectro electromagnético. Puede ser pasivo (recibe la potencia de activación a través de su antena) o activo (tiene su propia fuente de potencia, por ejemplo una batería). Las principales diferencias con la tecnología **NFC** es que esta es bidireccional y funciona a frecuencias y distancias mucho más bajas, mientras que **RFID** es unidireccional, puede utilizar mayores frecuencias y, por ende, mayores distancias. Un ejemplo de sistemas que utilizan esta tecnología es un receptor para abrir una puerta de garaje. Por tanto, se entiende que es crítico que este sistema no sea vulnerable, ya que puede suponer un riesgo crítico.

3.3.5. Auditoría física

Por último, en el **ataque físico** se trata de desensamblar el disposi tivo e investigar su contenido.

- Se comienza con la **placa base** del dispositivo (**REQ-MA-5-1**). En ella se deben investigar el microcontrolador del dispositivo. De esta manera se puede descubrir más fácilmente si dispone de interfaces de desarrollador o **debug**. También conviene averiguar identificar las memorias RAM [25] y ROM [26] que utiliza, por si hiciera falta acceder al contenido de estas.

- A simple vista a la placa base se puede averiguar si el dispositivo dispone de alguna interfaz de desarrollador (**REQ-MA-5-2**). Esta interfaz se encuentra mediante una serie de pines colocados juntamente. Pueden ser tanto una interfaz de **comunicación serie** (**UART**) o de **debug**. La comunicación mediante el protocolo **UART** [7] (del inglés *Universal Asynchronous Receiver-Transmitter*, Emisor-Receptor Universal Asíncrono) se realiza transmitiendo los datos en serie desde el emisor hacia el receptor, que después convertirá los datos de la manera adecuada para su tratamiento posterior. Identificando esta interfaz se pueden leer todas las comunicaciones que se realicen en el dispositivo a través de este protocolo. Adicionalmente, sobre toda la placa también se pueden descubrir

puntos de prueba del dispositivo, aunque el foco principal son las interfaces mencionadas anteriormente.

- Continuando con la auditoría, en caso de no haber encontrado de ninguna manera el firmware de manera pública, se puede proceder a **extraer** este **firmware** de las **memorias del dispositivo** (**REQ-MA-5-3**). Se encuentra en la memoria ROM del dispositivo, y se debe **volcar** su contenido y tratarlo adecuadamente para poder analizarlo correctamente. Ello también permite averiguar si existe más contenido de datos almacenado en la memoria además del firmware como, por ejemplo, una configuración de usuario creada previamente por otro usuario el cual había restaurado los datos del dispositivo de fábrica.

- El dispositivo puede incluir diversos **sensores** para poder cumplir con su propósito, los cuales también será necesario analizar y auditar debidamente (**REQ-MA-5-4**). Estos sensores pueden comunicarse vía **UART** y, si se consigue encontrar una interfaz serie desde la cual leer las comunicaciones, se pueden analizar sus tramas (es decir, cómo se comunica el sensor con el microcontrolador) y determinar si estas comunicaciones son seguras.

- Por último, en caso que se consiga interactuar con la interfaz de desarrollo del dispositivo, se pueden analizar el **sistema de arranque** y el **sistema operativo** del dispositivo (**REQ-MA-5-5**). Desde el sistema de arranque se puede alterar el funcionamiento del dispositivo y conseguir acceso al control del dispositivo previo a la carga del sistema operativo. Con ello se pueden averiguar las configuraciones de inicio del dispositivo, lo cual puede ser crítico si se descubren datos sensibles como contraseñas.

3.3.6. Resumen de vulnerabilidades

Una vez finalizadas las tres auditorías, se procede a redactar un **resumen de todas las vulnerabilidades** encontradas en ellas (**REQ-MA-6-1**). Todas las vulnerabilidades deben tener el **proceso** que se ha seguido para encontrarlas, una **descripción** que indique en qué medida afecta al usuario final del dispositivo y la **gravedad** de la vulnerabilidad encontrada. Además, es prácticamente necesario añadir una **posible solución** o alternativa a cada vulnerabilidad. También es importante mencionar todo aquello que **no se pudo auditar**, ya que de cara a la redacción del informe final será necesario para informar al cliente de todo aquello que no se ha podido o no ha dado tiempo a auditar.

3.3.7. Redacción del informe final

Por último, una vez puntuadas y catalogadas las vulnerabilidades, se debe realizar un **resumen ejecutivo** de todo el proceso de la auditoría (**REQ-MA-7-1**). Este debe incluir un resumen de las **vulnerabilidades** encontradas, **ordenadas** desde las más graves hasta las que menos, incluyendo secciones meramente informativas para el cliente. Este resumen pertenecerá al informe final que será completado por la empresa y entregado por su parte al cliente que ha solicitado la auditoría.

3.4. Matriz de trazabilidad de requisitos

La matriz de trazabilidad enlaza los requisitos de usuario junto con los requisitos de la metodología de auditoría. Esta matriz viene representada en la Tabla 3.29.

Requisitos		Requisitos de usuario (**REQ-U-XX**)						
		U-1	**U-2**	**U-3**	**U-4**	**U-5**	**U-6**	**U-7**
Requisitos de la metodología (**REQ-MA-XX**)	**MA-1-1**	**X**						
	MA-1-2	**X**						
	MA-1-3	**X**						
	MA-1-4	**X**						
	MA-2-1		**X**					
	MA-2-2		**X**					
	MA-2-3		**X**					
	MA-3-1			**X**				
	MA-3-2			**X**				
	MA-3-3			**X**				
	MA-4-1				**X**			
	MA-4-2				**X**			
	MA-4-3				**X**			
	MA-4-4				**X**			
	MA-5-1					**X**		
	MA-5-2					**X**		
	MA-5-3					**X**		
	MA-5-4					**X**		
	MA-5-5					**X**		
	MA-6-1						**X**	
	MA-7-1							**X**

Tabla 3.29: Matriz de trazabilidad del proyecto.

Capítulo 4

Aplicación de la metodología de auditoría

En este capítulo se aplicará la metodología de auditoría desarrollada sobre **dos dispositivos** IoT: una **cerradura inteligente** y un **dispositivo PLC**. En un principio, se documentaría la aplicación de esta metodología a la **aplicación móvil** auditada durante la estancia en prácticas, pero debido a contratos de confidencialidad no será posible.

4.1. Auditoría de cerradura inteligente

En esta sección se tratará de auditar una cerradura inteligente de una manera básica, siguiendo la metodología planteada para obtener el máximo de puntos vulnerables de este dispositivo. Como la metodología indica, se resumirá todo lo encontrado en el apartado del resumen.

4.1.1. Identificación del dispositivo

Antes de comenzar la auditoría, hay que identificar el propósito del dispositivo y qué protocolos utiliza para cumplir con su función. El propósito de este dispositivo es el de ofrecer métodos tecnológicos para abrir una cerradura, ya sea para una casa, un hotel, etc. Los métodos que utiliza la cerradura auditada son: el protocolo **NFC**, un sistema de **huella dactilar** y una **aplicación móvil** (y por tanto, conexión **WiFi** o **Bluetooth**). También utiliza medios más tradicionales, como un panel numérico o una llave física para abrir la cerradura. Se investiga además el interior de la cerradura, encontrando lo que podrían ser varias interfaces de **serie** o **debug** en la placa base del dispositivo.

4.1.2. Definición del vector de ataque

Una vez realizado el proceso de identificación del dispositivo, se anotan todos los puntos de vulnerabilidades encontrados. Se ha encontrado lo siguiente:

- Se puede analizar y auditar el firmware del dispositivo.
- Se ha encontrado una aplicación móvil genérica que funciona con la cerradura.
- Al tener aplicación móvil, debe tener conexión a Internet, por tanto se puede auditar los protocolos de red y los puertos que tenga en uso.
- Se puede analizar y auditar el protocolo NFC de la cerradura, principalmente de sus tarjetas.
- Se puede investigar si tiene página de configuración.
- Se puede auditar el sensor de huella dactilar.
- Se pueden analizar los pines encontrados en la placa base. en búsqueda de interfaces **serie** o **debug**.
- En caso de encontrar una interfaz serie o debug, auditar el sistema de arranque del dispositivo.

Una vez desglosados todos los puntos, se clasifican en su respectivo vector de ataque. En cuanto al vector de **ataque remoto**, solamente se clasificará el análisis del **firmware** del dispositivo. La **aplicación móvil** que utiliza y los **protocolos y comunicaciones de red** hacia el exterior que realiza el dispositivo se pueden auditar en el ataque remoto. No obstante, es más recomendable realizar estas auditorías en el **ataque local**. De esta manera se conseguirán mejores análisis de las vulnerabilidades. Además, para continuar con el ataque local, se puede auditar el protocolo **NFC**. Finalmente, en relación al **ataque físico**, se pueden analizar y auditar los **puertos serie** o **debug** encontrados en el análisis rápido realizado a la placa base del dispositivo. Seguido de ello, se puede analizar y auditar el sensor de **huella dactilar** y sus tramas de comunicación con la cerradura.

4.1.3. Auditoría remota

En la auditoría remota se puede auditar y analizar el **firmware** del dispositivo, la **aplicación móvil** que utiliza y las **comunicaciones de red** que se realicen. Pero, debido al tiempo disponible para auditar la cerradura, no hubo tiempo para auditar el firmware. Además, como se ha comentado anteriormente, las auditorías de la **aplicación móvil** y de las **comunicaciones de red** hacia el exterior se pueden realizar de mejor manera en el **ataque local**.

4.1.4. Auditoría local

Dentro de la auditoría local, se encuentran diversos puntos de ataque. Se puede analizar el protocolo **NFC**, la **aplicación móvil** y las **comunicaciones de red** del dispositivo:

Para poder realizar la auditoría al protocolo **NFC**, se utiliza la siguiente guía [48]. La cerradura utiliza la aplicación genérica para autorizar el acceso a las tarjetas **NFC** de esta. Para poder analizar estas tarjetas, se pueden utilizar las aplicaciones móviles **NFC Tools** [59] y **Mifare Classic Tools** [57]. Con la primera aplicación se puede averiguar cuál es la versión de tarjetas **Mifare** que se utilizan para la cerradura. En este caso, son las **Mifare Classic 1K**. Con la segunda aplicación se pueden leer los sectores y la información almacenada en estos,

como se ven en las Figuras 4.1 y 4.2. Si se analizan los datos encontrados y, usando como guía la leyenda de colores de la aplicación, se puede ver que en el primer sector aparece la información del fabricante y la UID (del inglés *User Identification*, Identificación de Usuario) de la tarjeta. También se puede ver que el resto de sectores son completamente visibles y legibles con todo su contenido con el valor *0*. Esto indica que esos sectores están vacíos y no están cifrados. Si los sectores estuvieran cifrados, aparecería un error a la hora de leer dichos sectores y no aparecerían en el resultado final.

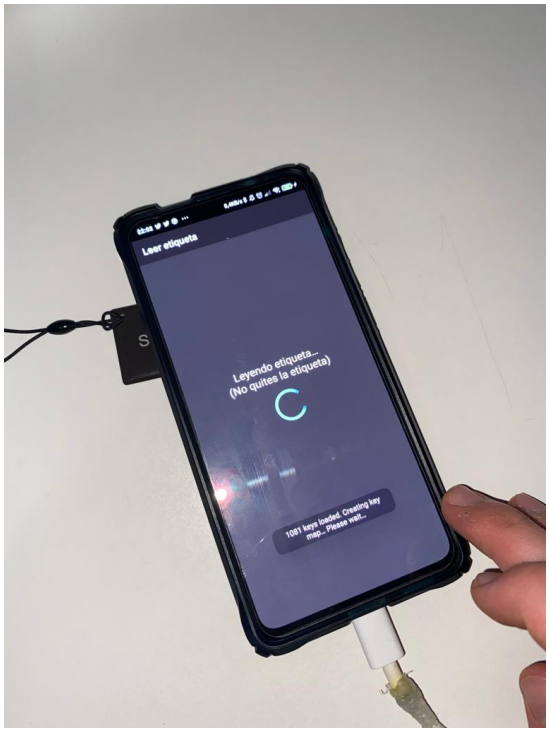

Figura 4.1: Escaneo de los sectores de la tarjeta.

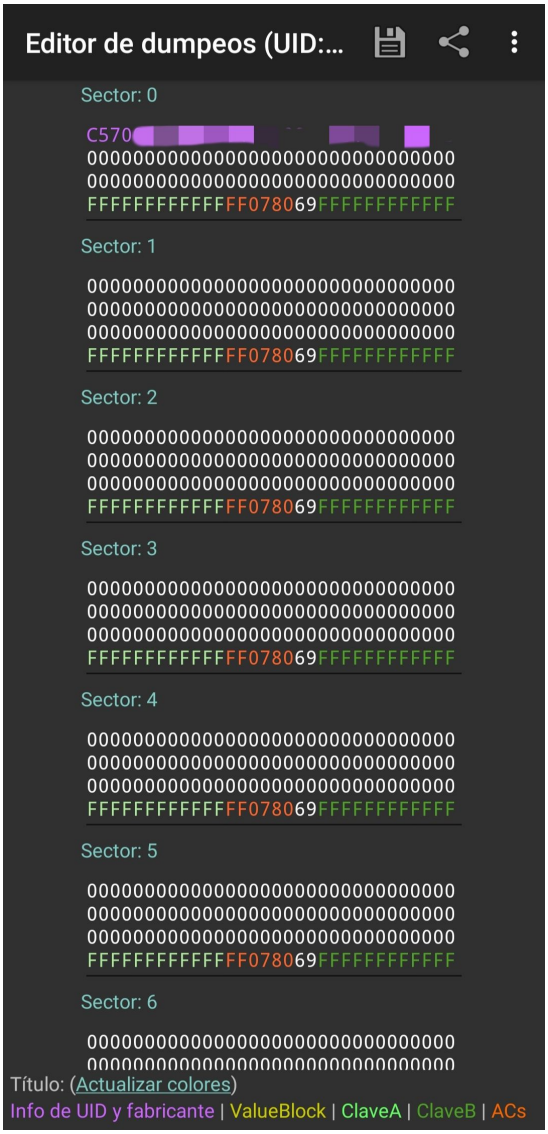

Figura 4.2: Resultado del escaneo con los datos sensibles censurados.

Por tanto, si los sectores están vacíos y sin cifrar y solo tiene datos escritos el primer sector, se puede deducir que la verificación que tiene la cerradura para identificar que la tarjeta es una tarjeta autorizada para poder abrir la cerradura es mediante la UID de la tarjeta. Esto es un fallo grave debido a que cualquier tarjeta que pueda clonar o duplicar la misma UID que la tarjeta original podría emular la autorización. Y, por tanto, abrir la cerradura sin problemas. Para solucionar este problema, se debería verificar de una manera más estricta esta autorización, utilizando los sectores vacíos para almacenar datos de verificación y cifrando dichos sectores.

Otro problema encontrado es en relación a la versión de las tarjetas, ya que son de tipo **Mi-**

fare Classic 1K. Si se visualiza de nuevo la Figura 4.1, se puede ver que aparecen marcados de distintos tonos de color verde dos claves: la **ClaveA** y la **ClaveB**. Existe una vulnerabilidad en relación a estas que, mediante estas claves, permiten la vulneración de cualquier clave utilizada para esta generación de tarjetas, utilizando herramientas como **mfcuk** [16], **mfoc** [17] o **libnfc** [15]. También se podría utilizar la herramienta de **miLazyCracker** [18]. Con lo cual, cifrar los sectores no sería suficiente. Se tendrían que actualizar a la siguiente generación de tarjetas, las **Mifare Classic 4K**, que son más seguras.

De esta manera se da por finalizada la auditoría del protocolo **NFC**. A continuación se prosigue con la auditoría de la **aplicación móvil** que utiliza la cerradura.

De manera general, las aplicaciones móviles también pueden tener vulnerabilidades importantes. Para preparar la auditoría de la aplicación móvil se ha seguido la siguiente guía [36] y se han instalado los siguientes programas o programas de comandos:

- NoxPlayer: emulador de móviles Android con posibilidad de activación de modo root
- Burp Suite Community Edition: capturar y analizar tráfico de red [45]
- Frida [8] y Objection [29]: software libre para poder deshabilitar comprobación de certificados a la hora de capturar el tráfico de red
- JADX: para decompilar las APKs de Android
- Android Studio y adb: para poder analizar las APKs decompiladas y extraer los datos de ella

También es muy recomendable tener un dispositivo móvil Android con **root** activado, es decir, con permisos de control total sobre el dispositivo móvil. Como no se disponía de él, se ha utilizado la máquina virtual **NOX Player** con **root** activado. Una vez listo, se instala la aplicación por auditar en la máquina virtual y se comienza la utilización de la aplicación, lo que requiere de registro de usuario. Se siguen los pasos necesarios para ello y se utiliza de manera básica para que la aplicación almacene algunos datos para analizarlos más adelante, como por ejemplo claves de usuario para abrir la cerradura o números pin. Una vez realizados estos pasos, se toman tres vías:

La primera es decompilar la **APK** de la aplicación (del inglés *Android Application Package*, Paquete de Aplicación de Android) utilizando el programa **JADX**. Una vez decompilada, junto con el programa **Android Studio**, se pueden visualizar todos los archivos de la aplicación. El primero que se investiga es el archivo **"AndroidManifest.xml"** (en adeltante, **Manifest**). Un ejemplo de este archivo se ve en la Figura 4.3). En este archivo se pueden comprobar todos los permisos que necesita la aplicación, además de información extra. También se pueden visualizar todos los archivos **"java"** que dispone la aplicación para que esta funcione correctamente. Para auditarla, se deben buscar en todos los archivos cadenas de datos escritas a mano (en inglés, *hardcoded strings*) que hagan relación a la encriptación de los datos de la aplicación, por ejemplo, buscar la sigla **AES**, (del inglés *Advanced Encryption Standard*, Estándar de Encriptación Avanzada). Se puede buscar en el archivo **Manifest** si está configurado el permiso de **debug** o de **backup**. Estas dos opciones deberían estar siempre deshabilitadas para versiones de la aplicación lanzadas en producción y es un fallo de seguridad dejarlas habilitadas. En cuanto al parámetro de **debug**, dejarlo habilitado supone ofrecer la posibilidad a que todo usuario que sepa utilizar los permisos de administrador pueda vulnerar la aplicación. Y en cuanto al

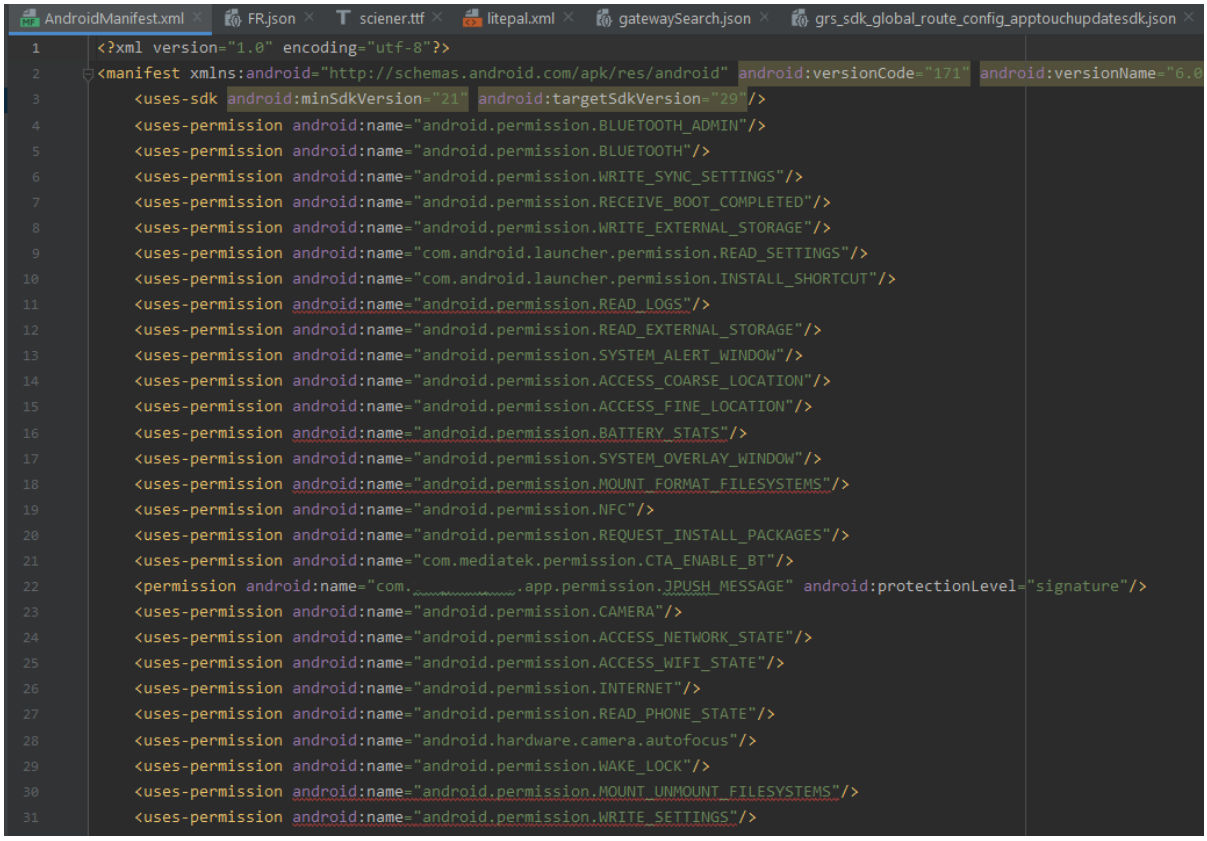

Figura 4.3: Información encontrada en el archivo **AndroidManifest.xml**.

parámetro **backup**, este permite al usuario realizar copias de seguridad de la aplicación desde un ordenador, sin requerir autorización alguna [1].

En la segunda vía se tratará de investigar todos los archivos y datos almacenados por la aplicación, como se ve en la Figura 4.4. Esto son, por ejemplo, archivos de **caché**, archivos **xml** o **bases de datos** donde se almacenen datos como la cuenta del usuario, el correo, su contraseña o números de identificación (IDs) de sesión, entre otros. Se investigan todos los archivos y ficheros que se han conseguido extraer y se anotan todos los aquellos que puedan contener datos de carácter sensible: datos cifrados, datos en claro o las tablas de bases de datos donde se almacenan esos datos (aunque estén vacías). El uso previo de la aplicación permite encontrar datos de usuario y datos cifrados en los archivos analizados. Una de las vulnerabilidades encontradas en esta auditoría es que la clave de encriptación **AES** se encuentra en claro en la base de datos. Conociendo esta clave, la cual es la clave privada de encriptación de los datos, se pueden descifrar todos los datos que se almacenen en la aplicación que estén cifrados con esta clave. Esta clave privada debería estar cifrada.

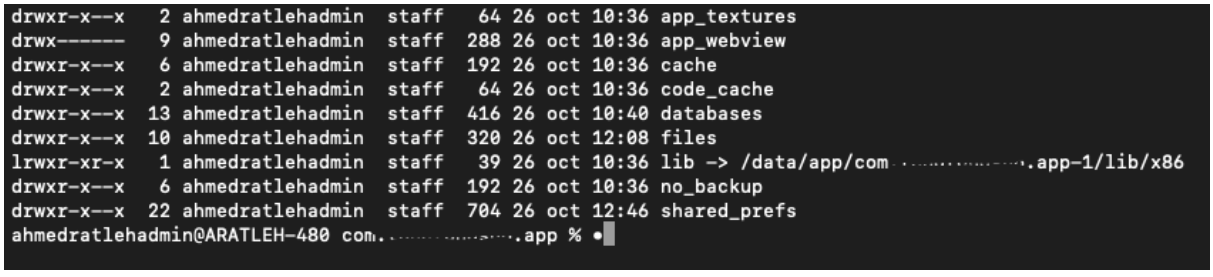

Figura 4.4: Directorios extraídos al decompilar la aplicación.

Finalmente, la tercera vía consta de capturar el tráfico de la aplicación, comprobar su seguridad y visualizar su contenido. De esta manera también se audita el **tráfico de red** mencionado anteriormente. Esta aplicación utiliza verificación por certificados, con lo cual no es tan sencillo capturar el tráfico y que funcione correctamente. Primero, hay que configurar el móvil de la máquina virtual y el programa **Burpsuite** para que todo el tráfico de red que salga del móvil emulado tenga que enrutarse a través de **BurpSuite** mediante un **Proxy** [24] (servidor intermediario que trata los datos desde una posición intermedia entre el usuario y la dirección donde se está intentando comunicar, como una página web). Una vez hecho, para disponer de conexión a Internet sin problemas hace falta instalar en el móvil emulado un certificado, ofrecido por **BurpSuite** [44]. Después de ello se comprueba si se dispone de conexión a Internet y que no hayan problemas con ella. Debido a la verificación por certificados, además de la instalación del certificado, hay que evitar esa verificación utilizando programas de comandos como **Objection** y/o **Frida**. Una vez activado uno de los dos, se prueba a utilizar la aplicación y se comprueba que ya funcione con normalidad. Se comprueba que el tráfico se captura de manera correcta desde el programa **Burpsuite**. El tráfico capturado ayuda a comprobar que no se transfieren datos de carácter sensible, a través de la red, a un servidor remoto totalmente desconocido. También se comprueba que solamente se envíen los necesarios para poder hacer comprobaciones de conexiones. En la Figura 4.5 se visualiza un ejemplo del tráfico capturado. Se puede ver la ID de la **cookie** [28] en claro, cuando es más recomendado que esté cifrada.

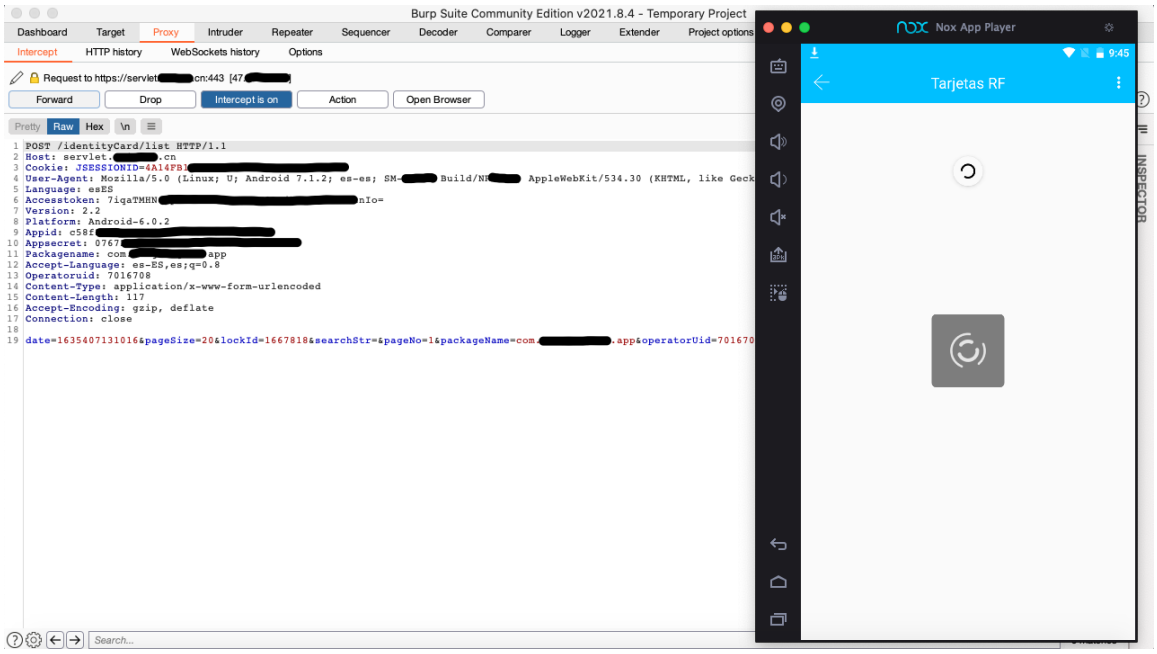

Figura 4.5: Información encontrada a la hora de capturar tráfico de red.

4.1.5. Auditoría física

Para comenzar la **auditoría física**, se realiza un análisis del hardware del dispositivo, identificando los componentes que tiene en su placa base. Se utilizan las figuras Figura 4.6 y 4.7 como referencia para explicar las secciones de una placa base de la cerradura inteligente.

La sección coloreada de verde en las dos figuras representa las posibles interfaces **serie** o **debug** del dispositivo. Mediante un análisis y seguimiento de las pistas de los puertos de estas interfaces se puede deducir si se utilizarán para alguno de los propósitos mencionados anteriormente. Para ello, hay que identificar el microcontrolador. Este se muestra en la sección

coloreada de azul. En este caso se encuentran tres posibles microcontroladores adicionales en la placa, aunque también pueden ser otros elementos. Generalmente, estos elementos tienen escrito su nombre de componente en la parte de arriba junto con una marca para indicar su orientación, que siempre se colocará en la esquina superior izquierda del componente (véase la Figura 4.8 y su sección coloreada de blanco). Por tanto se puede intentar leer e identificar qué tipo de componente es, averiguar los pines de salida que tiene (en inglés, **pinout**) y con qué se conectan.

Todo dispositivo electrónico necesita de una señal de reloj que sincronice todos los elementos de la placa junto con el microcontrolador. En la sección de color rojo de la Figura 4.6 se encuentran diversos relojes de cristal de cuarzo, encargados de generar las señales de reloj necesarias para el funcionamiento del microcontrolador y del dispositivo.

Las secciones coloreadas de naranja y amarillo de la Figura 4.6 representan el controlador de **NFC** de la cerradura y un puerto de antena, respectivamente. En la Figura 4.7, en la sección coloreada de negro, se encuentra un botón. En este caso, es un botón de **reset** (en español, botón de reinicio de fábrica). Finalmente, se identifica que el resto de componentes son resistencias, condensadores o chips de puertas lógicas, entre otros, encargados de la correcta comunicación y alimentación de los componentes.

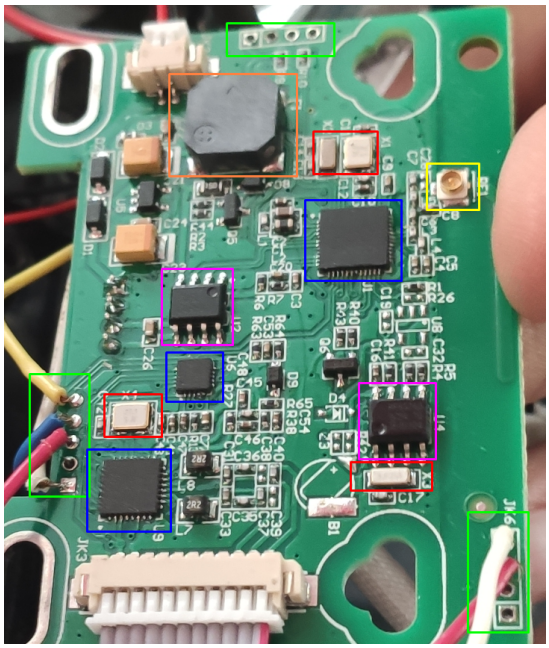

Figura 4.6: Cara delantera de la PCB del dispositivo.

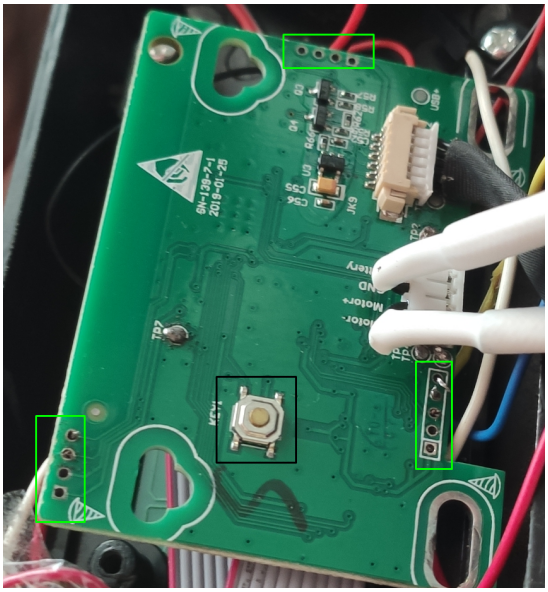

Figura 4.7: Cara trasera de la PCB del dispositivo.

Para finalizar la auditoría física, se procede a auditar el protocolo de la huella dactilar. Para ello, se puede utilizar la **interfaz serie** encontrada del dispositivo para comprobar si el tráfico de la cerradura se realiza por comunicación serie. Para ello se realiza la conexión mostrada en las figuras Figura 4.10, 4.11 y 4.12. Se han soldado dos cables en los puertos serie para poder conectarlos a un pequeño analizador lógico de 24 MHz (Figura 4.9) y un cable en un pin de **tierra** para cerrar el circuito. El cable de tierra se conecta al pin **GND** del analizador lógico, mientras que los otros dos cables se conectan a cualquier pin de canales que estén disponibles. En este montaje se han utilizado los canales 1 y 3. De esta manera, el analizador lógico está conectado vía USB al ordenador.

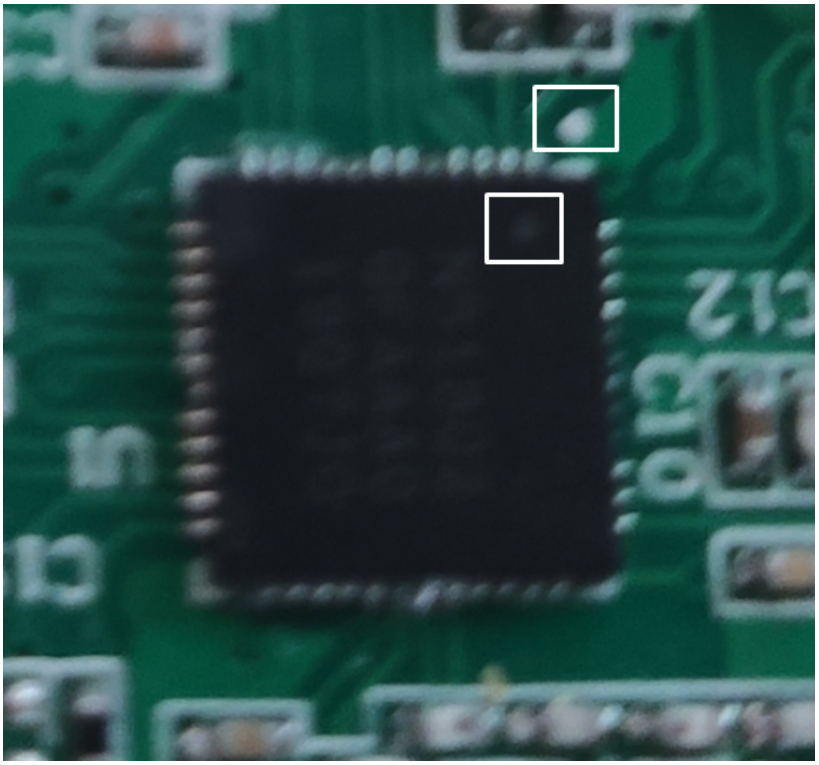

Figura 4.8: Texto escrito en el microcontrolador del dispositivo y sus marcas identificadoras.

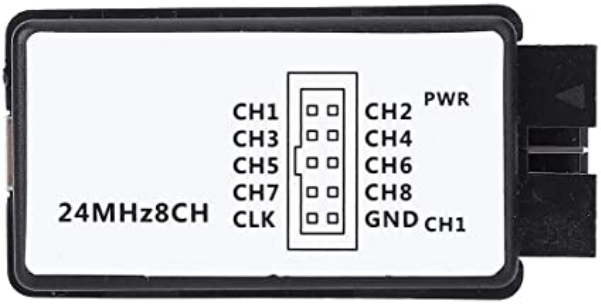

Figura 4.9: Configuración de pines del analizador lógico utilizado.

Una vez realizado el montaje, se analiza el tráfico del puerto serie mediante un programa llamado **Logic Analyzer**, de Saleae [47]. Una vez realizado, se utiliza la cerradura como si se fuera a acceder vía huella dactilar. Se descubre que el sensor de la huella dactilar sí que se comunica vía serie, con lo cual se puede analizar la trama del protocolo que transmite. Las tramas analizadas son: trama de inicialización, de huella aceptada, de huella incorrecta y de huella no reconocida. Estas tramas se pueden investigar en Internet para encontrar coincidencias. Realizando esto se descubren los sensores que posiblemente utiliza esta cerradura: el sensor **R307** [30], el sensor **R502** [31], el sensor **ZFM-70** [61], el sensor **GT-511C2** [2] o el sensor de **SparkFun Electronics** [50]. Los más parecidos (o con más coincidencias) son los dos primeros, **R307** y **R502**. Investigando más a fondo, se descubre que este sensor almacena la huella dactilar en el propio sensor. Para comunicar el acierto de la huella, el sensor le comunica a la cerradura la identificación de la página donde se encuentra la huella. Cuando esto ocurre, la cerradura se abre. Analizando los datos mostrados en la Figura 4.13, dónde se han realizado tres tramas de acierto de huella, se puede ver que los campos escritos en rojo de los mensajes marcados en rojo

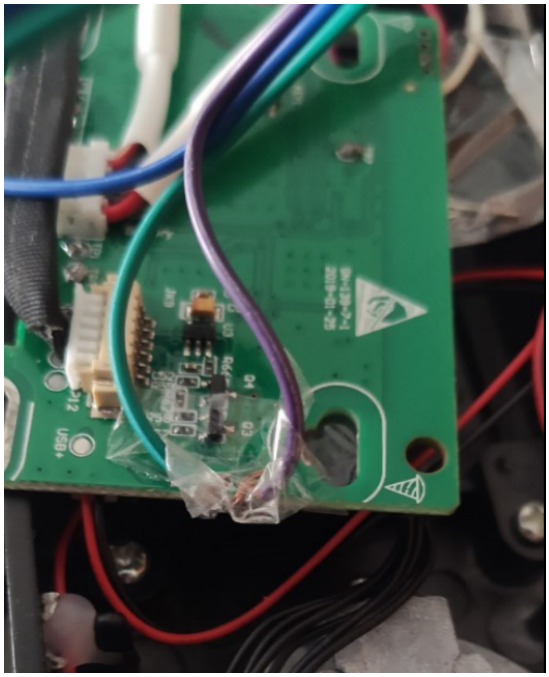

Figura 4.10: Cables soldados a la **interfaz serie** de la cerradura.

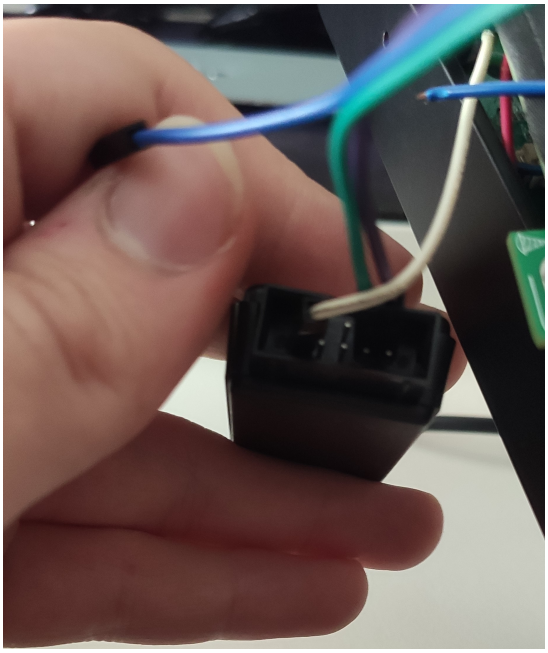

Figura 4.11: Cables conectados al analizador lógico.

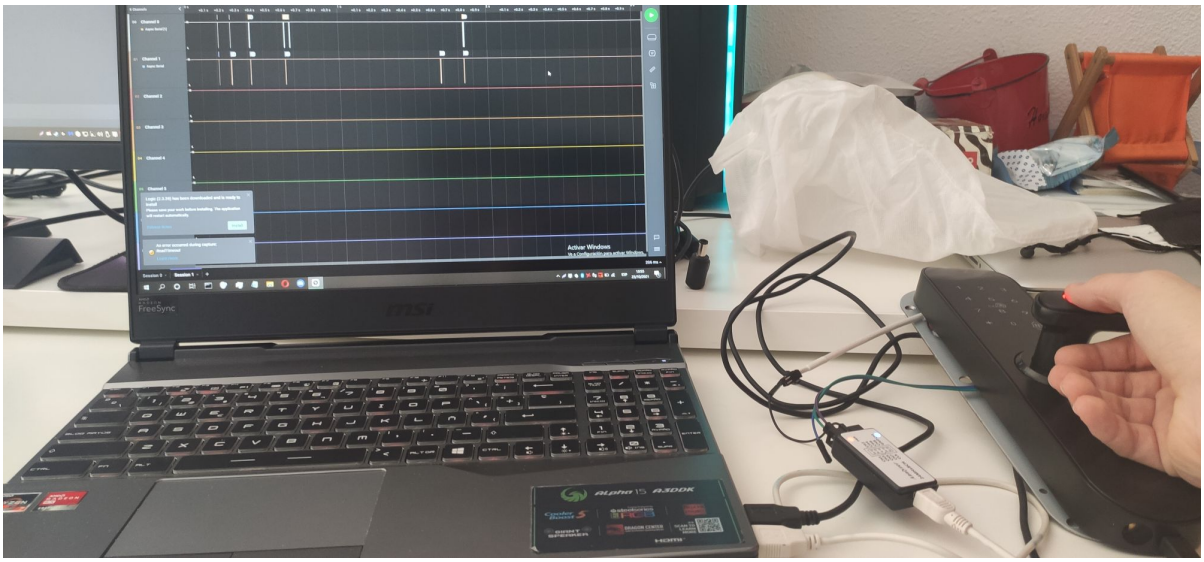

Figura 4.12: Montaje para el análisis de las tramas de la huella dactilar.

son siempre iguales. Esta es la identificación de la página que se transmite desde el sensor a la cerradura. Esto implica que existe la posibilidad de emular falsamente la comunicación, claro está, de manera física.

Figura 4.13: Tramas analizadas de las tramas de la huella dactilar.

Una manera de solucionar este problema es que, tanto el sensor como la cerradura (o cualquier dispositivo que utilice protocolos de reconocimiento de huella dactilar) verifiquen de una manera más segura la semejanza o igualdad de la huella introducida. Por ejemplo, en lugar de enviar la identificación de la página, que se envíen los datos de la huella cifrados y se comprueben directamente desde la cerradura, y no desde el sensor.

4.1.6. Resumen de vulnerabilidades

Después de realizar los tres focos de ataque hacia el dispositivo, se resumen y se puntúan las vulnerabilidades encontradas cada una en su categoría. Estas puntuaciones se realizan mediante la calculadora de puntuación [40] del **Instituto Nacional de Estándares y Tecnología**, donde según la puntuación obtenida, se clasifica la severidad de las vulnerabilidades en: ninguna (0 puntos), baja (de 0.1 a 3.9 puntos), media (de 4.0 a 6.9 puntos), alta (de 7.0 a 8.9 puntos) y crítica (de 9.0 a 10.0 puntos) [39].

- **Vulnerabilidades encontradas en la auditoría remota**: En relación al **ataque remoto**, no se han podido analizar otros puntos de ataque que contengan vulnerabilidades.
- **Vulnerabilidades encontradas en la auditoría local**: En cuanto al **ataque local**, se han encontrado diversas vulnerabilidades. La primera de ellas es la baja seguridad de las tarjetas **NFC** que se utilizan con la cerradura. Estas tarjetas operan con la tarjeta solo mediante un número identificador, sin comprobaciones de seguridad extra. Además, ningún sector se encuentra cifrado (debido a que no los utiliza). Aunque no sería de mucha utilidad, debido a que la versión de las tarjetas que utiliza contienen otra vulnerabilidad que permite obtener las claves de cifrado mediante las claves públicas que tiene la tarjeta. Por tanto, esta vulnerabilidad tiene una **puntuación** de **5.7**, con un riesgo **medio**. Otra vulnerabilidad encontrada es, en el análisis de la aplicación genérica de la cerradura, se ha encontrado la clave privada **AES** de cifrado de datos en claro. Esto permite que cualquier cifrado que utilice esta clave no sea útil, ya que se puede descifrar los datos mediante esta clave privada. Con lo cual, esta vulnerabilidad obtiene una **puntuación** de **6.5**, clasificándose en un nivel de riesgo **medio**.
- **Vulnerabilidades encontradas en la auditoría física**: Las vulnerabilidades encontradas en el ataque físico se relacionan con la interfaz serie que se ha encontrado. La primera es haber encontrado varias interfaces de pines de prueba o **testpoints**. Una de esas interfaces es de comunicación serie. Tener estas interfaces habilitadas en la fase de producción es una vulnerabilidad. Por tanto, según las puntuaciones de la página citada anteriormente, se obtiene una **puntuación** de **3.5**, con un riesgo **bajo**. Además, gracias a esta interfaz, se han podido analizar las tramas del sensor de la **huella dactilar**. De esta manera, se ha descubierto que este sensor realiza comprobaciones únicamente mediante identificación de paginación (similar al proceso de **NFC**). Por tanto, al no disponer de un protocolo más robusto y seguro de verificación de huellas dactilares, esta vulnerabilidad tiene una **puntuación** de **4.8**, catalogándose de riesgo **medio**.

Una vez redactadas todas las vulnerabilidades encontradas, se mencionan todos los puntos de ataque que no se han podido auditar. Estos puntos son los siguientes:

- Se puede investigar y analizar el firmware del dispositivo.
- En caso de no encontrar el firmware público, se puede volcar el contenido de la memoria.
- Si no se pudiera volcar el contenido de la memoria desde la interfaz serie, se puede extraer la memoria y volcar su contenido de manera física.
- Se puede auditar el sistema de arranque del dispositivo.
- Se puede investigar si tiene una página de configuración, además de la aplicación, y auditarla.

4.1.7. Redacción del informe final

Por último, se redacta el resumen ejecutivo de la auditoría. En este resumen se recopilan todas las vulnerabilidades encontradas, ordenadas desde las más graves (según la puntuación realizada) hasta las que menos, incluyendo los puntos meramente informativas. Una manera de recopilar estas vulnerabilidades es mediante una tabla, como la Tabla 4.1.

Código	Vulnerabilidad	Riesgo	Puntuación	Vector
VL-01	Clave privada AES descubierta en claro en los archivos de la aplicación	Medio	6.5	Local
VL-02	Protocolo NFC inseguro	Medio	5.7	Local
VF-01	Protocolo de huella dactilar inseguro	Medio	4.8	Físico
VF-02	Interfaz serie habilitada en producción	Bajo	3.5	Físico

Tabla 4.1: Resumen de vulnerabilidades encontradas en la auditoría de la cerradura.

Las **soluciones** a estas vulnerabilidades son las siguientes:

- Para la vulnerabilidad **VL-01**: No incluir la clave AES dentro de los archivos de la aplicación. En caso de hacerlo, se debe cifrar y guardar de manera que ningún usuario pueda acceder a ella, excepto los desarrolladores.
- Para la vulnerabilidad **VL-02**: Actualizar la versión de las tarjetas Mifare, utilizar mejores validaciones de las tarjetas, incluyendo más información en el resto de sectores para comprobar la veracidad de la tarjeta, y cifrar dichos sectores.
- Para la vulnerabilidad **VF-01**: Cambiar de sensor (debido a que el sensor en sí es inseguro) y utilizar mejores protocolos de autenticación, como por ejemplo, que la cerradura (y no el sensor) compruebe la igualdad de la huella almacenada y la que se ha introducido utilizando métodos más robustos que una identificación.
- Para la vulnerabilidad **VF-02**: Desarrollar un diseño de la placa base que no ofrezca una interfaz serie (o debug) para el dispositivo en fase de producción.

De esta manera concluye la parte del auditor en la redacción del informe final de la auditoría de la cerradura inteligente.

4.2. Auditoría de un dispositivo PLC (Power Line Communications)

En esta sección se tratará de auditar un dispositivo PLC de una manera básica, pero siempre abarcando el máximo de vías donde ser puedan encontrar vulnerabilidades. Todo ello se indicará en el resumen de las vulnerabilidades.

4.2.1. Identificación del dispositivo

Al ser un dispositivo de red se debe analizar el tráfico de red que realiza, sea vía WiFi o vía red eléctrica, junto con los puertos que utiliza o que tenga abiertos. Investigando en los manuales de usuario del dispositivo se encuentra que dispone tanto de aplicación de escritorio, de aplicación móvil como de una página web que se utiliza para la gestión del dispositivo. También se ha encontrado el firmware del dispositivo de manera pública, por lo tanto no hace falta extraerlo de las memorias del dispositivo. Como el dispositivo no dispone de más funciones adicionales ni más tecnologías, se investiga el interior del dispositivo, encontrando a simple vista lo que podría ser una interfaz serie.

4.2.2. Definición del vector de ataque

Después de realizar el proceso de identificación del dispositivo, se tiene que definir el vector de ataque con todas las secciones que se puedan auditar. El resultado para este dispositivo en concreto es el siguiente:

- Se puede analizar el tráfico red de manera remota y local
- Se pueden analizar los puertos de red que utiliza
- Dispone de aplicación de escritorio, de aplicación móvil y de aplicación web para configurar los parámetros del dispositivo
- El firmware del dispositivo es de descarga pública.
- Se encuentra disponible una interfaz de pines en la placa base del dispositivo.

Se tienen que clasificar estos puntos en su sección de ataque correspondiente. Para el **ataque remoto**, se analiza el firmware del dispositivo, las comunicaciones hacia Internet y las aplicaciones que dispone a excepción de la página web, ya que esta requiere de acceso al dispositivo. Respecto al **ataque local**, se analizará todo lo relacionado con el tráfico de red que genere. También se analizarán la aplicación web generada por el dispositivo para gestionar sus configuraciones. Por último, respecto al **ataque físico**, se tratará de identificar todos los componentes del dispositivo y determinar qué utilidad tienen los pines encontrados a primera vista anteriormente.

4.2.3. Auditoría remota

En esta sección se tratará de analizar el firmware del dispositivo y de encontrar fallos de seguridad en el. Para ello, se han seguido las guías [42], [35] y [21], ofrecidas por el supervisor. Ya que es de descarga pública, se descarga desde la página del fabricante la versión más actual posible y se procede con su análisis utilizando el programa de comandos **Binwalk**. Con él se obtienen los datos mostrados en la Figura 4.14.

Figura 4.14: Información obtenida al ejecutar **binwalk** con el archivo firmware.

Con la información obtenida se consiguen los siguientes datos:

- Tiene un procesador MIPS
- Tiene un sistema operativo Linux embedido (basado en OpenWrt)
- Su sistema de ficheros es squashfs
- La dirección de memoria de inicio del sistema de ficheros es 0x160000

Una vez averiguado qué sistema de ficheros utiliza y qué sistema operativo es, se puede probar a descomprimir el sistema de ficheros mediante el programa de comandos **Unsquashfs**. Pero antes, hay que averiguar si el archivo del firmware es un archivo abierto, comprimido o cifrado. Se puede averiguar utilizando la herramienta de la Figura 4.15, la cual muestra una gráfica de entropía. Una gráfica de entropía similar a la resultante indica que el archivo firmware solamente está comprimido. Si no estuviera comprimido, la gráfica mostrada no sería tan lineal, sino que estaría más distorsionada. En la gráfica resultante se obtienen caídas en la linealidad, lo cual indica que el firmware no está cifrado. En caso de estarlo, sería totalmente lineal con minúsculas variaciones.

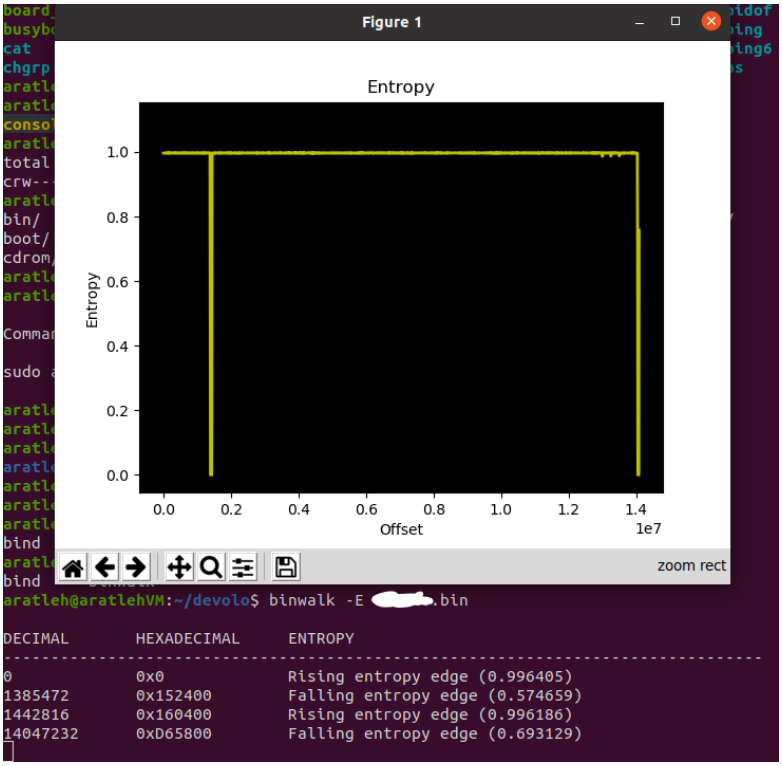

Figura 4.15: Entropía del archivo firmware.

Al averiguar que el firmware solamente está comprimido, se procede a su descompresión. Para hacer este proceso más rápido y fácil, se puede reducir el tamaño del firmware a descomprimir hasta la dirección de inicio del sistema de ficheros, la cual es **0x160000**, descubierta anteriormente. Para reducirlo se utiliza el comando que se muestra en la Figura 4.16. El número **1441792** es la dirección de memoria mencionada pero en decimal, ya que el comando necesita la versión decimal de la dirección. Después de esta reducción, y sabiendo que su sistema de ficheros es de tipo **squashfs**, se procede a la descompresión con el comando **unsquashfs**. Como se ve en la Figura 4.17, se ha generado una carpeta root del firmware.

El firmware ya está descomprimido y listo para ser analizado. Hay diversas maneras de

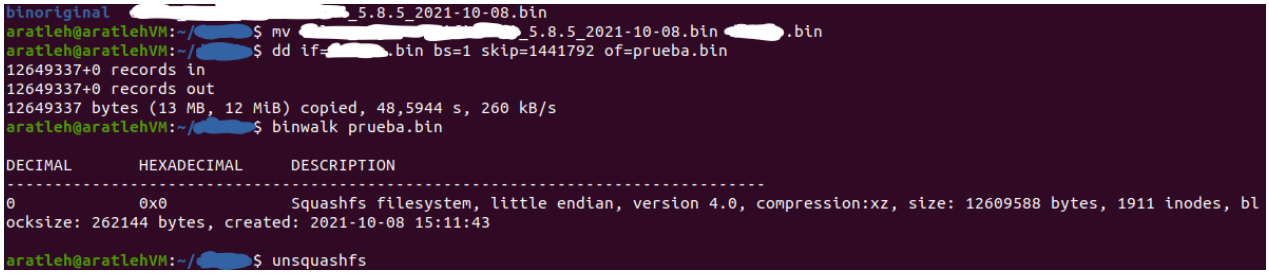

Figura 4.16: Creación del archivo firmware reducido, **prueba.bin**.

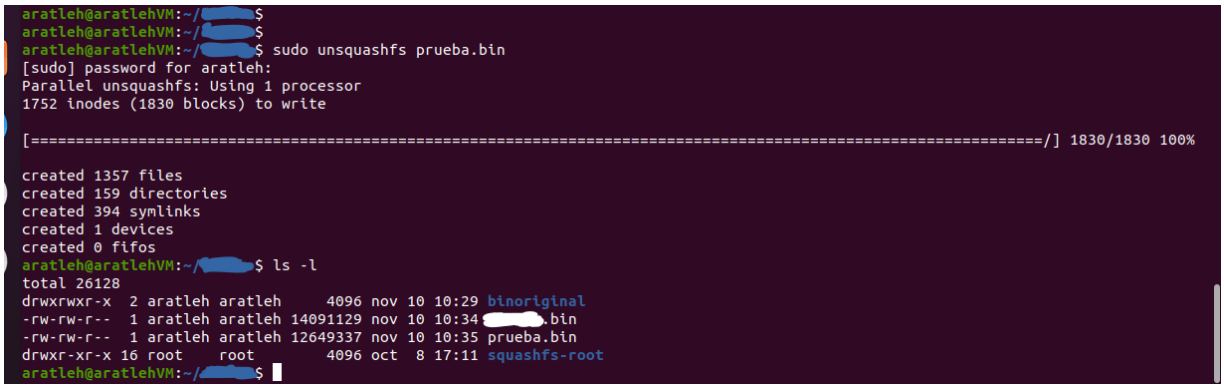

Figura 4.17: Sistema de ficheros descomprimido.

comenzar el análisis buscando puntos de ataque, en este caso se comenzará comprobando los archivos **shadow** [12] y **passwd** [11]. Estos archivos están localizados en el directorio **/etc**. En la Figura 4.18 se pueden visualizar los datos de estos ficheros. Analizando estos, se obtiene nueva información:

- Como se basa en OpenWrt, dispone de un usuario Avahi
- No aparecen las contraseñas de los usuarios en los archivos
- Aparecen distintos carácteres en los campos de las contraseñas en los dos archivos

En los dos archivos, el primer campo es el nombre de usuario y el segundo campo (separado por el carácter **":"**) es la contraseña de ese usuario. El archivo **passwd** muestra con el carácter **"X"** el campo de la contraseña cuando esta se muestra cifrada en el archivo **shadow**. Si se muestra con el carácter **"*"**, significa que no se puede iniciar sesión con esa cuenta. El archivo **shadow** muestra el usuario **root** con la contraseña vacía. Esto significa que se puede entrar en el usuario root sin aplicar contraseña.

4.2.4. Auditoría local

Para abordar la **auditoría local**, se realizarán el **escaneo** y **análisis** de **puertos de red** y el **tráfico de red** del dispositivo. Para poder escanear los puertos del dispositivo, hay que establecer conexión interna con el dispositivo, sea vía **Ethernet** o vía **WiFi**. En la Figura 4.19 se puede ver cómo está conectado vía cable. De esta manera, averiguando la **dirección IP**

```
aratleh@aratlehVM:~/[redacted]/squashfs-root/etc$
aratleh@aratlehVM:~/[redacted]/squashfs-root/etc$
aratleh@aratlehVM:~/[redacted]/squashfs-root/etc$ sudo cat passwd
root:x:0:0:root:/root:/bin/ash
daemon:*:1:1:daemon:/var:/bin/false
ftp:*:55:55:ftp:/home/ftp:/bin/false
network:*:101:101:network:/var:/bin/false
nobody:*:65534:65534:nobody:/var:/bin/false
avahi:x:105:105:avahi:/var/run/avahi:/bin/false
ntp:x:123:123:ntp:/var/run/ntp:/bin/false
aratleh@aratlehVM:~/[redacted]/squashfs-root/etc$
aratleh@aratlehVM:~/[redacted]/squashfs-root/etc$
aratleh@aratlehVM:~/[redacted]/squashfs-root/etc$ sudo cat shadow
root::0:0:99999:7:::
daemon:*:0:0:99999:7:::
ftp:*:0:0:99999:7:::
network:*:0:0:99999:7:::
nobody:*:0:0:99999:7:::
avahi:x:0:0:99999:7:::
ntp:x:0:0:99999:7:::
aratleh@aratlehVM:~/[redacted]/squashfs-root/etc$
```

Figura 4.18: Contenido de los archivos **shadow** y **passwd**.

del dispositivo, se puede utilizar el software de comandos **nmap** para escanear los puertos abiertos [43]. Hay dos tipos de protocolos de puertos, **TCP** [22] (del inglés *Transmission Control Protocol*, Protocolo de Control de Transmisión) y **UDP** [23] (del inglés *User Datagram Protocol*, Protocolo de Datagramas de Usuario). Mediante los comandos que se pueden ver en las figuras Figura 4.20 y 4.21 (los cuales escanean los puertos **UDP** y **TCP** de una **dirección IP** dada, respectivamente), se obtienen los puertos que el dispositivo tiene abiertos.

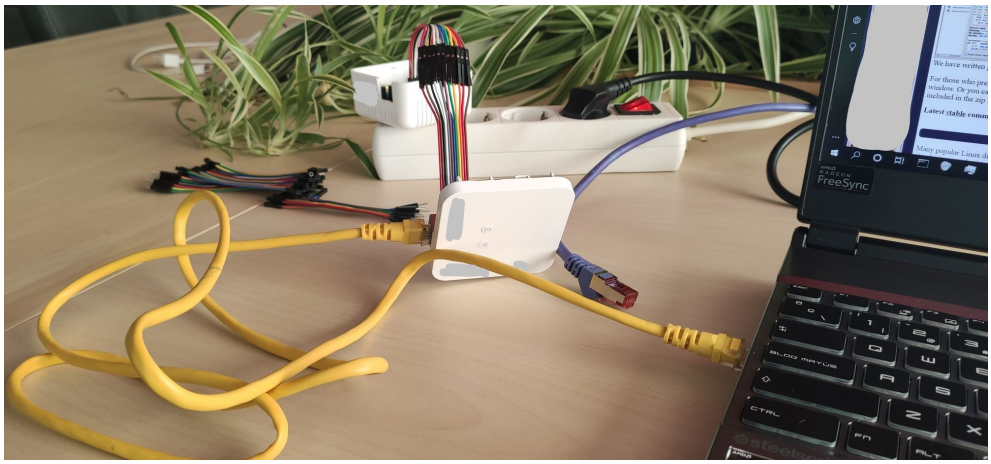

Figura 4.19: Conexión con el dispositivo vía cable **Ethernet**.

Analizando los resultados obtenidos en las figuras anteriores, se han encontrado los siguientes puertos abiertos, mientras que los demás están cerrados:

- TCP 80: puerto para HTTP
- TCP 443: puerto para HTTPS
- TCP 14791: puerto desconocido

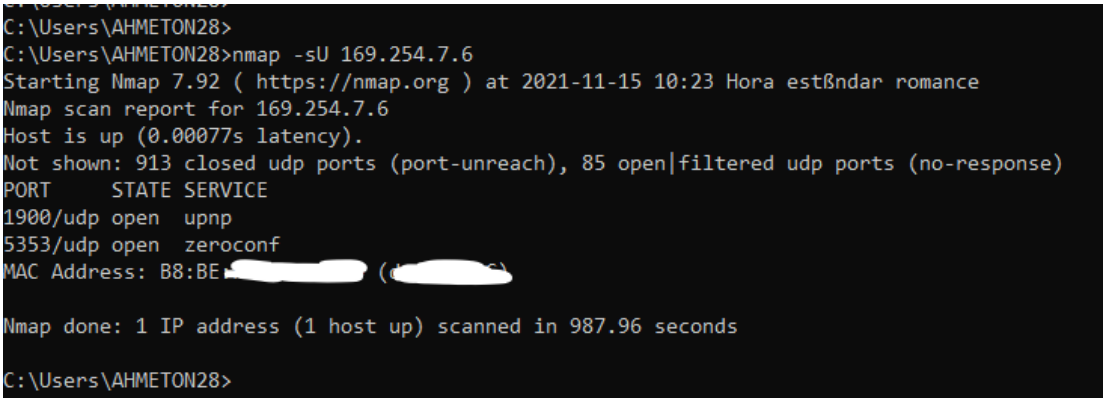

Figura 4.20: Puertos **UDP** abiertos.

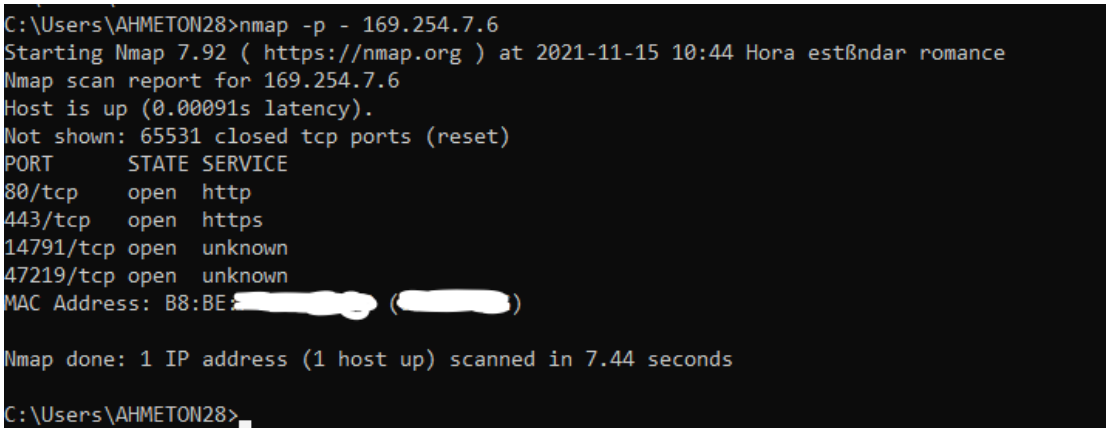

Figura 4.21: Puertos **TCP** abiertos.

- TCP 47219: puerto desconocido
- UDP 1900: puerto SSDP
- UDP 5353: puerto mDNS

Una vez identificados, es hora de analizar qué son y cuál es su uso. El PLC dispone de una página web para configurar los parámetros del dispositivo. Se sabe que es un dispositivo de red, por tanto, es natural que los **puertos 80** y **443** estén abiertos. Se tendrá que analizar si el tráfico que circula por ellos se realiza de manera segura o no. Esto es, que todo el tráfico circule principalmente por el puerto 443, ya que es el puerto de tráfico de red con protocolos de seguridad punto a punto. Los **puertos 14971** y **47219** son dos puertos desconocidos propios del PLC, por tanto hay que analizar qué tráfico se realiza a través de ellos y, de esta manera, descubrir su propósito. Respecto a los dos puertos UDP, sirven para las primeras configuraciones del PLC. El **puerto 1900** sirve para poder reconocer distintos dispositivos de la misma marca en la red para poder ser configurados entre sí, mientras que el **puerto 5353** está configurado por la distribución Avahi de Linux, que sirve para poder resolver los nombres de los diferentes dispositivos que puedan haber en la red.

El siguiente paso es averiguar el uso real de todos los puertos. Para ello primero se comenzará con los de tráfico de red para comprobar si se realiza de manera segura. Para analizarlo

se utilizará la herramienta **Wireshark** para poder capturar todo el tráfico posible. Además se interactuará con el dispositivo, por ejemplo, aplicando algunas configuraciones desde la aplicación de escritorio. De esta manera se puede averiguar si existe tráfico local. Se puede guardar un archivo que contiene todo el tráfico capturado para investigarlo sin la necesidad de permanecer conectado al dispositivo.

- El análisis comenzará con los **puertos 80 y 443**. Primero hay que definir que el **puerto 80** es el puerto destinado a las conexiones **HTTP** [58] (del inglés *Hypertext Transfer Protocol*, Protocolo de Transferencia de hipertexto) y el **puerto 443** está destinado a las conexiones **HTTPS** [52] (del inglés *Hypertext Transfer Protocol Secure*, Protocolo de Transferencia de Hipertexto Seguro), el cual es más seguro que el primero mencionado debido a los protocolos de seguridad que utiliza. Investigando el tráfico de red capturado, se ha descubierto que el **puerto 443** (**HTTPS**) no está en uso y que todo el tráfico circula por el **puerto 80** (**HTTP**). Esto se puede averiguar filtrando desde **Wireshark** las comunicaciones en estos puertos de la manera que se ven en las Figuras 4.22 y 4.23. Con lo cual, la primera de las vulnerabilidades encontradas es que todas las comunicaciones web circulan por el **puerto 80** y no por el **puerto 443**. Esta vulnerabilidad es más grave cuando circulan datos en claro, como por ejemplo, la contraseña de red del dispositivo.

- En relación al puerto **UPD 1900** no se encuentra tráfico interesante de red, ya que este protocolo pertenece al servicio **Upnp**. Este servicio permite conexiones automáticas entre dispositivos. Pero, debido a que el PLC no se ha conectado a ningún otro dispositivo distinto ni ninguno otro se han conectado a él, no hay tráfico que analizar.

- En cuanto al puerto **UDP 5353**, realiza bastantes comunicaciones con un servicio propio. Esto es debido a que este puerto utiliza el protocolo **mDNS** [33], que sirve para poder obtener los datos de otros dispositivos de la misma marca y conocerse entre sí dentro de la misma red.

- Respecto a los dos puertos desconocidos, se ha conseguido capturar tráfico a través del **puerto 14791** pero no del **puerto 47219**, como se puede ver en las Figuras 4.25 y 4.24. No se ha encontrado tráfico en el segundo puerto debido a que este también se utiliza para la vinculación de otros dispositivos de la misma marca entre sí.

Aplicando el mismo método de filtros por puerto, se puede filtrar el tráfico por cadenas de texto deseadas. Por ejemplo, si se cambia la contraseña desde el panel de administración durante la captura de tráfico por una personalizada (en este caso la contraseña se ha cambiado a *leakedhaha*), se puede buscar esta contraseña como una cadena de texto en todo el archivo del programa **Wireshark** y comprobar si la contraseña se envía en claro, es decir, sin estar protegida o cifrada. Como se puede ver en la Figura 4.26, sí que se ha encontrado la contraseña en el filtro establecido, por tanto se deduce que sí que circulan datos en claro a través del tráfico de red, lo cual es una nueva vulnerabilidad crítica encontrada.

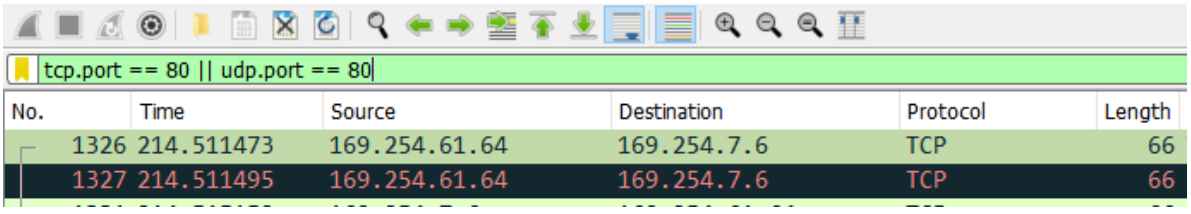

Figura 4.22: Se captura tráfico en el puerto **HTTP**.

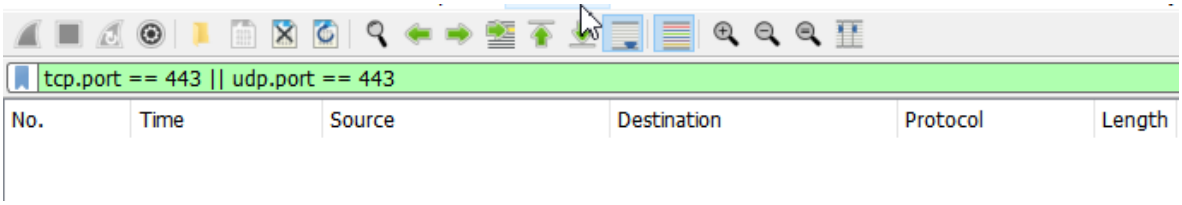

Figura 4.23: No se captura tráfico en el puerto **HTTPS**.

En relación a los puertos desconocidos, en relación al **puerto 47219** no hay datos al respecto, mientras que el **puerto 17791** realiza una petición **GET**, destacada en la Figura 4.25. Esta solicitud parece estar compuesta por un parámetro identificador, una ruta (censurada) y el nombre de la petición que está solicitando. Para seguir investigando esta petición, se prueba a buscar archivos que contengan esta petición

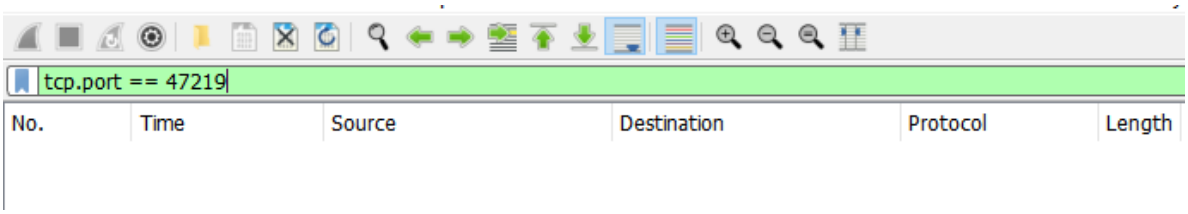

Figura 4.24: No se captura tráfico en el **puerto 47219**.

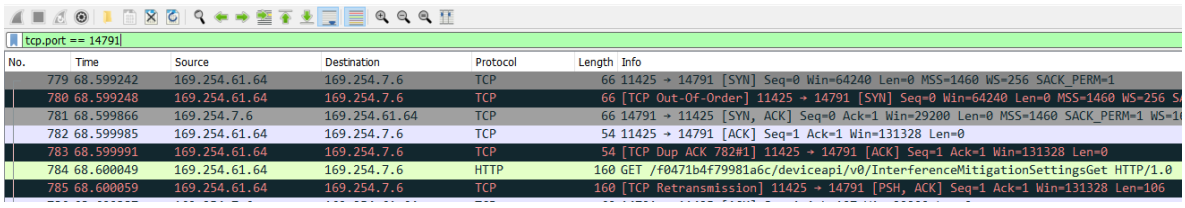

Figura 4.25: Se captura tráfico en el **puerto 14791**.

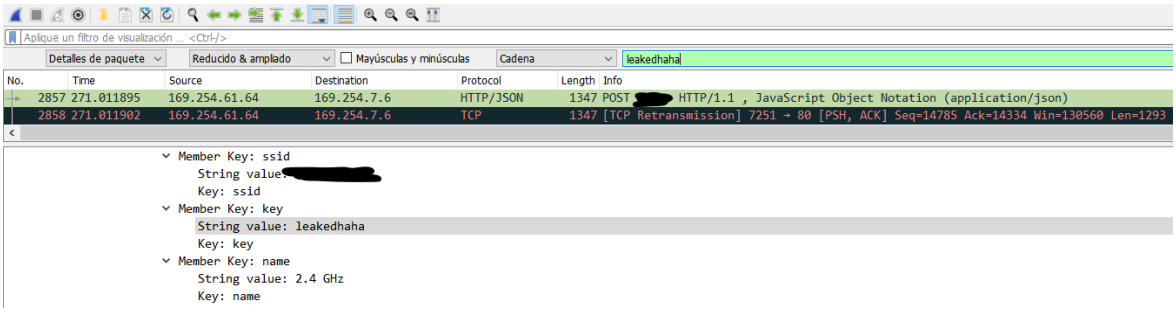

Figura 4.26: Contraseña filtrada a través del capturado de tráfico.

Para averiguar más información acerca de estos puertos, se puede intentar filtrar el número de estos en los **archivos del firmware** analizados en la **auditoría remota**. Para ello, se utiliza el comando de las Figuras 4.27 y 4.28. Se ha encontrado que existe un archivo dentro del firmware llamado **dlanApp2Backend** que contiene información en texto sobre estos puertos. Por tanto, se vuelve a realizar otra búsqueda dentro de los archivos del firmware, pero esta vez con el nombre recién encontrado, como se ve en la Figura 4.29. Como resultado, se encuentran dos archivos binarios, **dlanApp2Backend-device** y **dlanApp2Backend-plcnet**.

Como son archivos binarios se pueden desensamblar y analizar su contenido. En este caso no

```
aratleh@aratlehVM:~/[redacted]/squashfs-root$ grep -iRl "14791"
grep: dev/console: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-systemd-resolved.service-l71Tcf: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-switcheroo-control.service-qDskWi: Permission denied
grep: var/ssh-0HrkDmtKpHkQ/agent.1378: No such device or address
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-systemd-timesyncd.service-zua0bh: Permission denied
grep: var/.ICE-unix/1534: No such device or address
grep: var/.X11-unix/X0: No such device or address
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-fwupd.service-g28d9e: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-upower.service-98Ik1h: Permission denied
grep: var/snap.snap-store: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-ModemManager.service-Agqwzi: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-colord.service-OVE97f: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-systemd-logind.service-A6DjOh: Permission denied
grep: etc/resolv.conf: No such file or directory
etc/init.d/dlanApp2Backend
grep: etc/shadow: Permission denied
etc/rc.d/S99dlanApp2Backend
grep: etc/fstab: No such file or directory
grep: etc/TZ: No such file or directory
grep: www/cgi-bin/data-upload: No such file or directory
grep: www/cgi-bin/config-backup: No such file or directory
aratleh@aratlehVM:~/[redacted]/squashfs-root$
```

Figura 4.27: Resultado de filtrar el **puerto 14791** en los archivos del firmware.

```
aratleh@aratlehVM:~/[redacted]/squashfs-root$
aratleh@aratlehVM:~/[redacted]/squashfs-root$ grep -iRl '47219'
grep: dev/console: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-systemd-resolved.service-l71Tcf: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-switcheroo-control.service-qDskWi: Permission denied
grep: var/ssh-0HrkDmtKpHkQ/agent.1378: No such device or address
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-systemd-timesyncd.service-zua0bh: Permission denied
grep: var/.ICE-unix/1534: No such device or address
grep: var/.X11-unix/X0: No such device or address
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-fwupd.service-g28d9e: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-upower.service-98Ik1h: Permission denied
grep: var/snap.snap-store: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-ModemManager.service-Agqwzi: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-colord.service-OVE97f: Permission denied
grep: var/systemd-private-77f2250fd7c2436d9c90a3ebfae06d4e-systemd-logind.service-A6DjOh: Permission denied
grep: etc/resolv.conf: No such file or directory
etc/init.d/dlanApp2Backend
grep: etc/shadow: Permission denied
etc/rc.d/S99dlanApp2Backend
grep: etc/fstab: No such file or directory
grep: etc/TZ: No such file or directory
grep: www/cgi-bin/data-upload: No such file or directory
grep: www/cgi-bin/config-backup: No such file or directory
aratleh@aratlehVM:~/[redacted]/squashfs-root$
```

Figura 4.28: Resultado de filtrar el **puerto 47219** en los archivos del firmware.

hace falta utilizar un programa sofisticado de desensamblaje de archivos binarios como **GHidra**, es suficiente con aplicarle el comando **strings**. Este comando traduce los archivos binarios a textos más legibles por el ser humano. Después de visualizarlos una vez, se encuentra la siguiente información: la petición **GET** analizada anteriormente utiliza el método **InterferenceMitigationSettingsGet** (Figura 4.25). Este método se encuentra en el binario recién desensamblado, entre otros muchos métodos más. Esta petición comienza con la cadena **device.api**. Se puede aplicar un filtro de esta cadena sobre el archivo binario para obtener solamente los métodos más similares al capturado. En las Figuras 4.30 y 4.31 se recopilan los resultados de aplicar este filtro.

```
aratleh@aratlehVM:~/[redacted]/strings&bins$
aratleh@aratlehVM:~/[redacted]/strings&bins$ strings -n16 dlanApp2Backend-device | grep -w 'device.api.*'
device.api.FactoryResetStart
device.api.InterferenceMitigationProfileGet
device.api.InterferenceMitigationProfileSet
device.api.InterferenceMitigationProfileSetResponse
device.api.InterferenceMitigationSettingsGet
device.api.InterferenceMitigationSettingsSet
device.api.InterferenceMitigationSettingsSetResponse
```

Figura 4.30: Resultado de desensamblar el binario del **puerto 14791**.

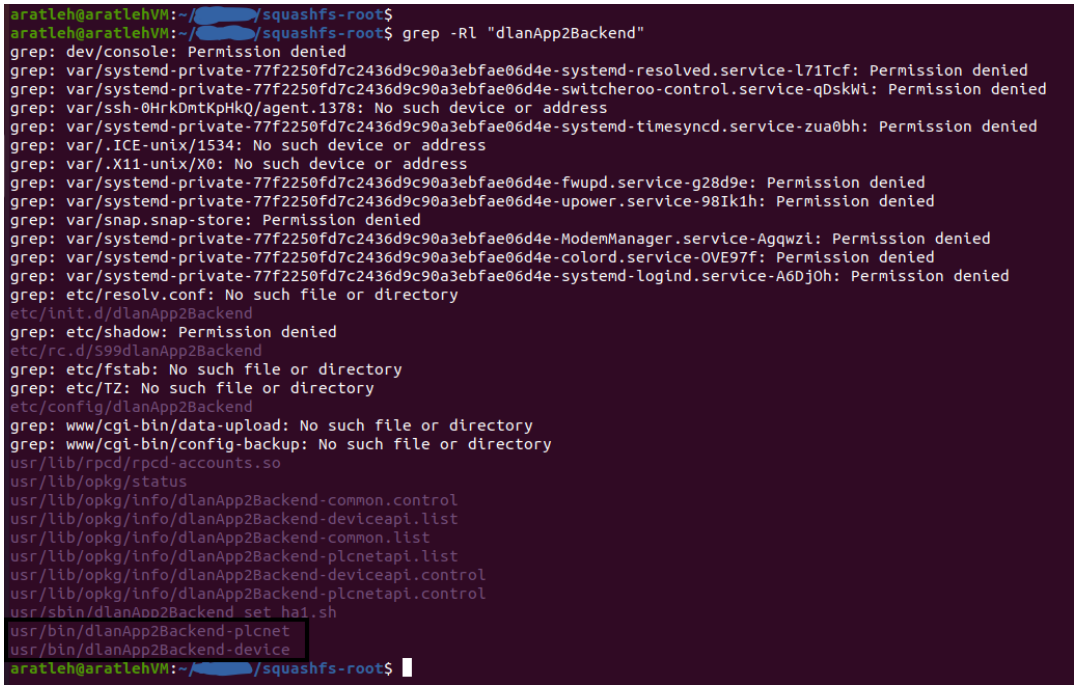

Figura 4.29: Resultado de filtrar el archivo encontrado en los demás archivos del firmware.

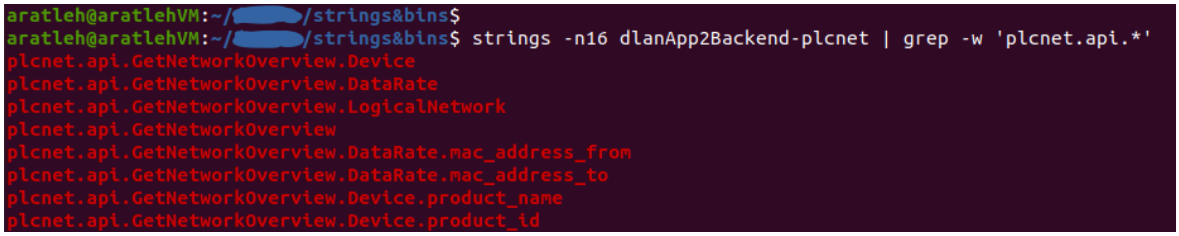

Figura 4.31: Resultado de desensamblar el binario del **puerto 47219**.

Viendo la cantidad de comandos que hay por explorar y explotar y habiendo adivinado la estructura de las peticiones, se prueba a emularlas. Para ello, se utiliza una herramienta llamada **netcat**. Se sigue la misma estructura que la que se visualiza en la Figura 4.32. Con lo cual, se prueba a realizar esta emulación con la instrucción capturada anteriormente, como se puede ver en la Figura 4.33.

```
nc -v 169.254.7.6 14791
GET /{id}/deviceapi/v0/{método deseado} HTTP/1.0
Host: 169.254.7.6:14791
“intro”
“intro”
```

Figura 4.32: Estructura de cómo emular las peticiones del dispositivo.

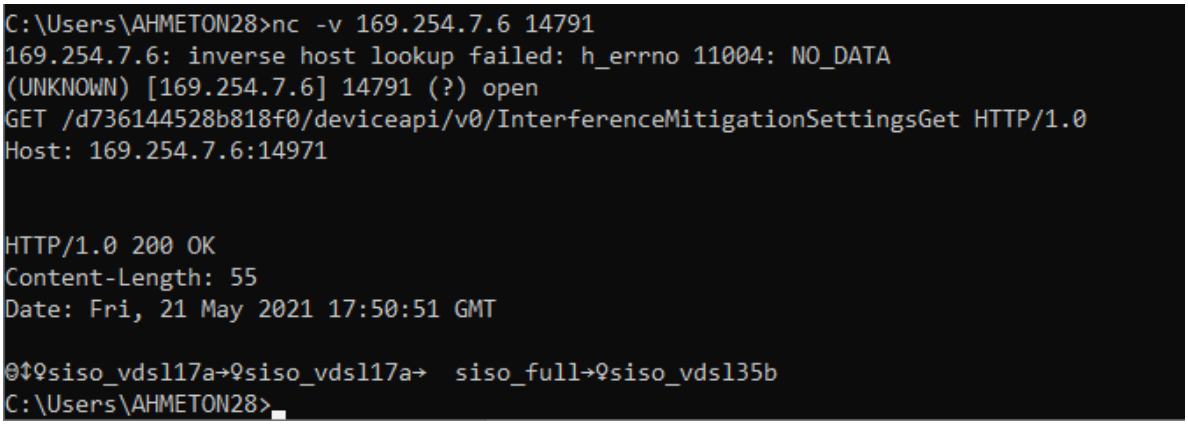

Figura 4.33: Proceso exitoso de emular la instrucción capturada anteriormente.

Como conclusión de este resultado, se averigua que la emulación de la instrucción ha funcionado, con lo cual se pueden realizar peticiones de manera general mediante **netcat**. Se prueba también a realizar una emulación de la instrucción de reinicio de fábrica del dispositivo (método **FactoryResetStart**), con resultado exitoso (véase la Figura 4.34). Esto es un problema grave (por ende, una vulnerabilidad), ya que acceder a funciones tan importantes de una manera tan sencilla no es seguro. Es necesario disponer de una autenticación segura para poder ejecutar comandos tan sensibles como restablecer el dispositivo de fábrica. En estas pruebas no se ha requerido de autenticación.

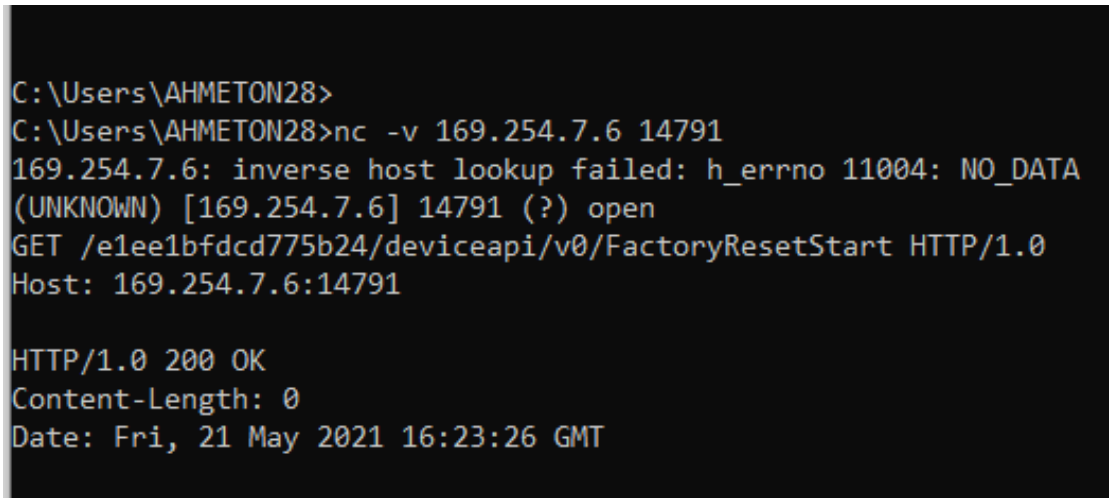

Figura 4.34: Proceso exitoso de emular la instrucción de reinicio de fábrica.

Para comprobar el problema de la autenticación, se accede a las configuraciones del dispositivo introduciendo la **dirección IP** en la barra superior de búsqueda. Cuando se accede a esta página de configuración, es imperativo que disponga de un usuario y una contraseña de acceso a la administración de estas configuraciones. Este dispositivo no dispone de un usuario de autenticación ni de contraseña de administración de fábrica, como se puede ver en la Figura 4.35. Con lo cual el usuario que utilice el dispositivo nunca accede a esta página y no cambia manualmente la contraseña, el panel de administración del dispositivo estará siempre abierto y desprotegido. Es por ello que las peticiones emuladas han funcionado, ya que si se introduce una contraseña, el resultado de la petición es fallida, dando error de autenticación necesaria (véase la Figura 4.36). No disponer de contraseña de administración con el dispositivo de fábrica es

una vulnerabilidad grave.

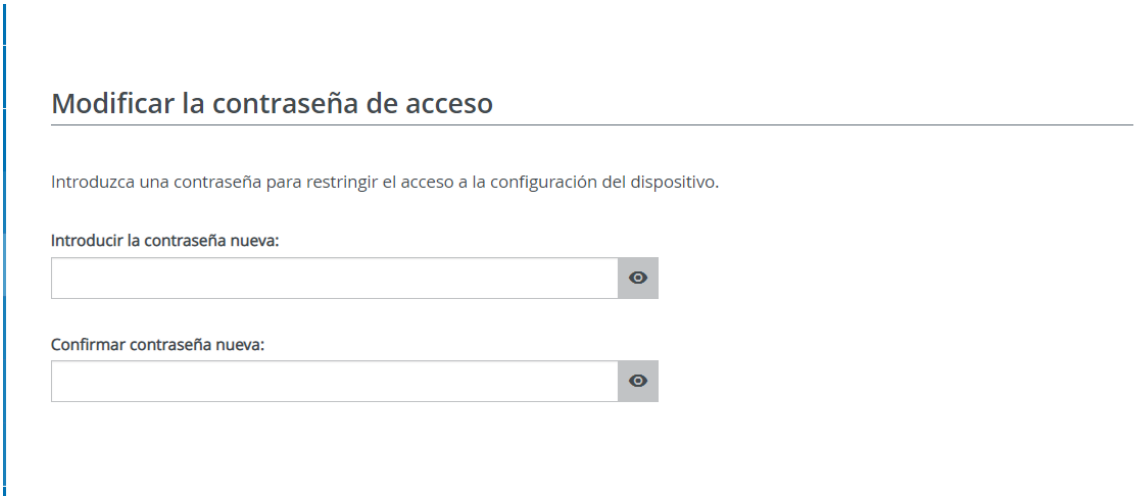

Figura 4.35: Campo de contraseña vacío con el dispositivo de fábrica.

Figura 4.36: Proceso fallido de emular la instrucción de reinicio de fábrica.

Investigando todos los posibles métodos que se pueden emular, se han encontrado varios que llaman la atención por su impacto en el funcionamiento del dispositivo o por los datos que se puedan filtrar:

- **FactoryResetStart**, reinicia de fábrica el dispositivo.
- **UpdateFirmwareStart**, comienza el proceso de actualización de firmware.
- **WifiParametersSet**, cambiar parámetros de la red del dispositivo.
- **WifiRepeaterParametersSet**, cambiar parámetros de la red WiFi repetida, generada por el dispositivo.
- **WifiGuestAccessGet**, obtener información de acceso a la red de invitados (nombre de red y su contraseña).
- **WifiWpsPbcStart**, activación del proceso **WPS** (del inglés *WiFi Protected Setup*, Configuración de WiFi Protegida).

Después de analizar los puertos y el tráfico de red, se procede a utilizar la página de configuración del dispositivo, siempre teniendo en cuenta que, inicialmente, esta no dispone de protección de fábrica. Ello permite que cualquier usuario, con o sin acceso al dispositivo, que

conozca la **dirección IP** predeterminada del dispositivo pueda entrar en el panel de configuración. Uno de esos paneles es la configuración de la red **WiFi**, donde puede modificar el nombre de la red, la contraseña y el canal utilizado como se ve en la Figura 4.37. Lo mismo ocurre con el panel de la red de invitados (Figura 4.38). Si se analizan bien estas dos figuras, se puede ver que la contraseña **por defecto** de la red de invitados, compuesta por 8 letras mayúsculas, es, de manera parcial, igual a la contraseña **por defecto** de la red principal, compuesta por 16 letras mayúsculas. Uno de los métodos antes mencionado se puede utilizar para obtener la información de la red de invitados (**WifiGuestAccessGet**), como se puede ver en la Figura 4.39.

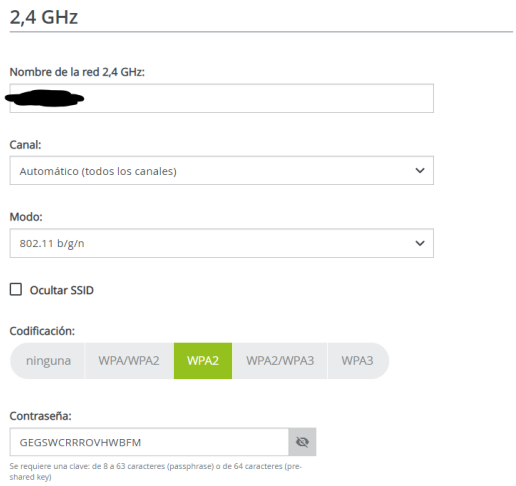

Figura 4.37: Panel de configuración de la red principal.

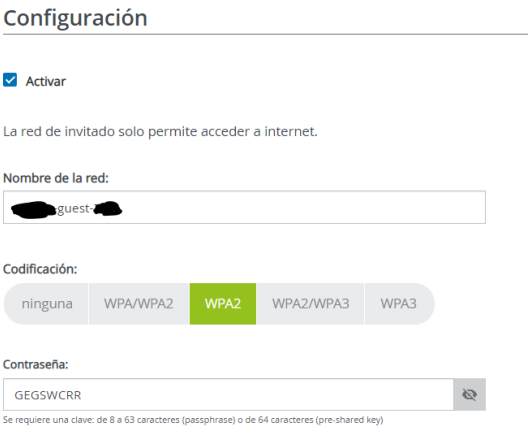

Figura 4.38: Panel de configuración de la red de invitados.

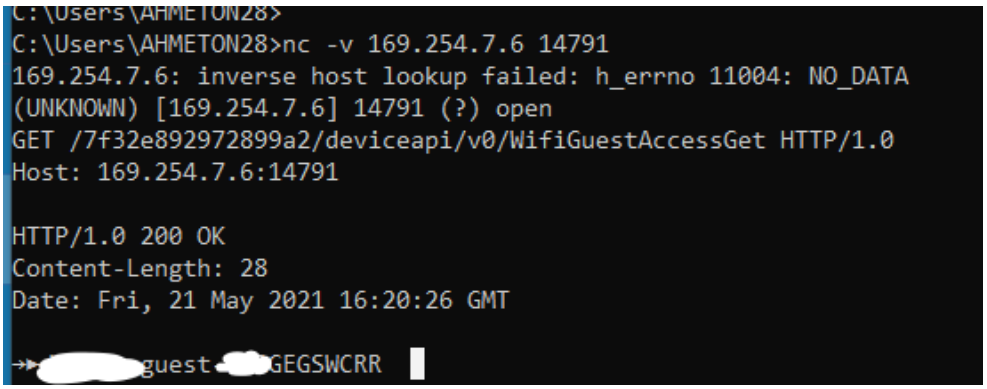

Figura 4.39: Información obtenida al emular el comando **WifiGuestAccessGet**.

Analizando la información obtenida con este comando, se observa que devuelve la contraseña de la red de invitados. Teniendo en cuenta que la contraseña obtenida es la primera mitad de la contraseña total del dispositivo, se podría emplear fuerza bruta para averiguar el resto de la contraseña. Utilizando esta página [56] se puede calcular el tiempo aproximado que tardaría en romper la contraseña con este método. Esta página indica el número total de combinaciones posibles, el tiempo en horas y en días que puede tardar en romper la contraseña y la cantidad de contraseñas probadas en una hora según la configuración de contraseña introducida previamente. También indica que los cálculos están basados en un procesador típico del año 2007 con una carga menor de su 10 %. Sabiendo que la contraseña por defecto de la red principal es de 16 letras mayúsculas y que la de la red de invitados es de 8 letras mayúsculas, se realizan estos cálculos en

la página citada anteriormente. Los resultados se pueden ver en las figuras Figura 4.40 y 4.41. Aunque parezca un tiempo infinito, se pueden utilizar tecnologías mucho más modernas para poder reducir ese tiempo a menos de un día, mediante el uso de tarjetas gráficas con capacidad de computación y cálculo muy superiores a las de los procesadores.

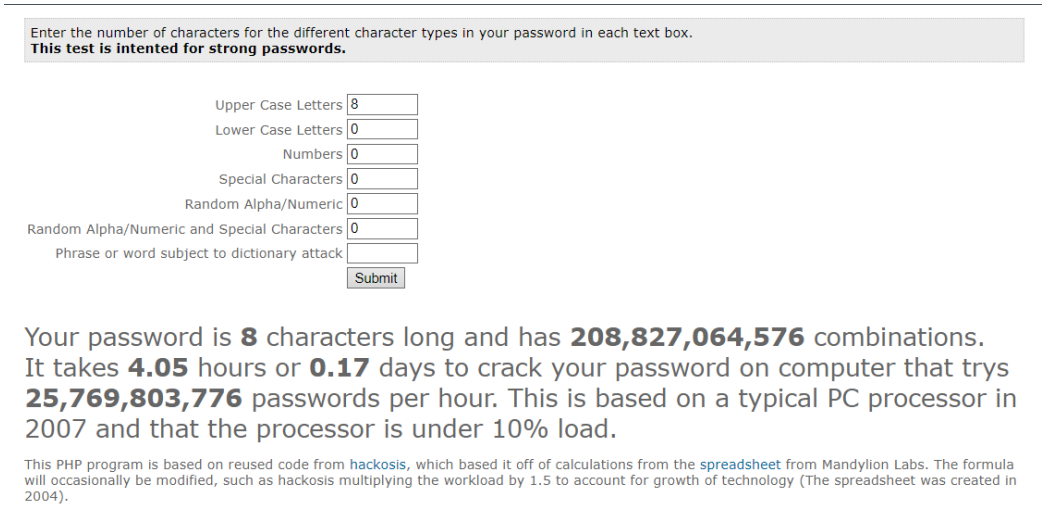

Figura 4.40: Tiempo aproximado que tardaría en romperse una contraseña de 8 caracteres alfabéticos, en mayúsculas y aleatorios.

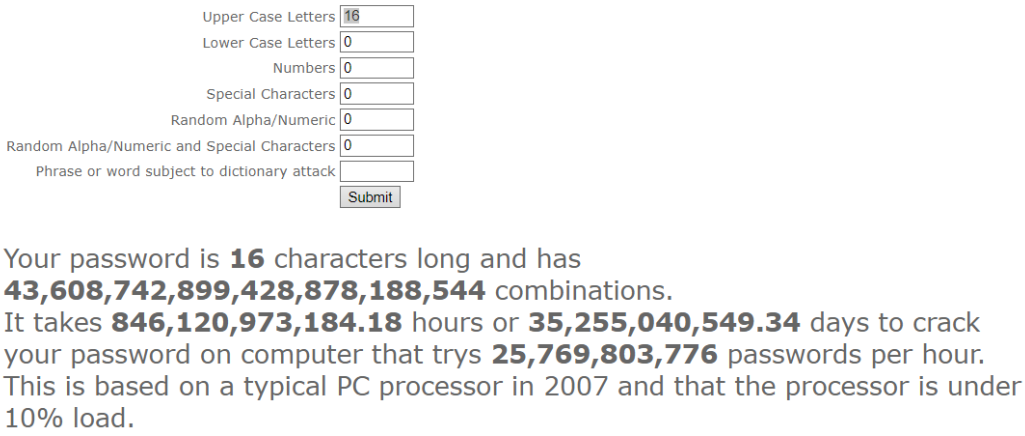

Figura 4.41: Tiempo aproximado que tardaría en romperse una contraseña de 16 caracteres alfabéticos, en mayúsculas y aleatorios.

Finalmente, se extrae una copia de seguridad de las configuraciones del dispositivo, desde el panel de administración, para analizar los datos que almacena y cómo los almacena. Para comprobar que realmente almacena datos que hayan sido modificados, se cambia la contraseña de la red principal del dispositivo, por ejemplo. Esta contraseña debería almacenarse cifrada, ya que sería una vulnerabilidad grave dejar la contraseña en claro. Pero, al analizar el archivo descargado del **backup**, se pueden ver todos los datos en claro, como se ve en las Figuras 4.42 y 4.43.

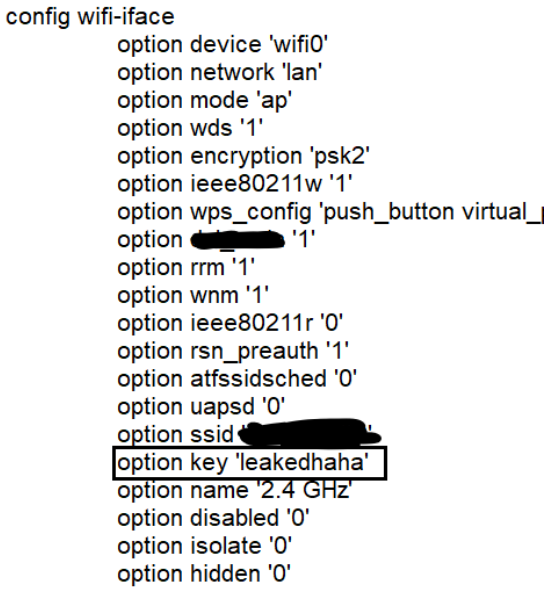

Figura 4.42: Información almacenada acerca de la red principal del dispositivo.

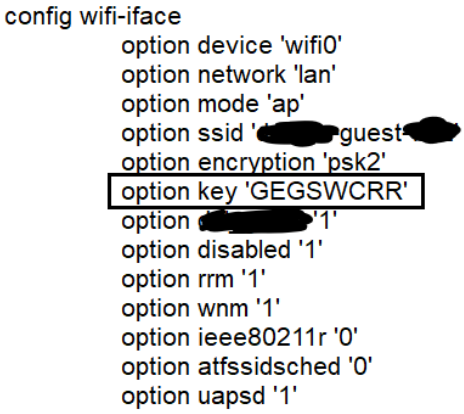

Figura 4.43: Información almacenada acerca de la red de invitados del dispositivo.

4.2.5. Auditoría física

El proceso de **ataque físico** se hace el último siempre, debido a que existe el riesgo de estropear el dispositivo. De esta manera se puede realizar todas las auditorías previas por completo antes de comprender este riesgo.

Para comenzar, hay que desensamblar el dispositivo por completo e identificar los componentes que hay en su placa base o PCB. En las figuras Figura 4.44 y 4.45 se pueden ver las dos caras de la placa del dispositivo con recuadros de colores identificando cada sección. En la primera figura se encuentra la cara delantera, donde están los botones que realizan las funciones básicas del dispositivo con sus luces LED (marcado en azul). Mientras que en la segunda hay un puerto de conexión de cable **Ethernet** (marcado en verde), una sección de pines hembra para conectarse con la otra mitad del dispositivo para su correcto funcionamiento (marcada en morado), una memoria **DRAM** (marcada en rojo) y una memoria **flash** (marcada en azul). En las dos caras se encuentra la interfaz de seis pines que se había visualizado al comienzo de la auditoría y una zona cubierta con una pequeña carcasa de metal. Si se quita la carcasa de la parte delantera, se encuentra el microcontrolador del dispositivo (marcado en rojo) junto con un material que le ayudará a disipar el calor generado (marcado en verde). Y, si se quita la carcasa de la cara trasera, se encuentran muchos componentes varios que forman parte del circuito del dispositivo y, al estar en la misma sección de la PCB que la carcasa delantera (Figura 4.47), del microcontrolador.

Lo importante de este análisis es el microcontrolador del dispositivo, la memoria flash que posee y la interfaz de pines al descubierto. En la Figura 4.46 se puede encontrar el nombre del microcontrolador, el cual si se busca en Internet se encuentra su hoja de datos o **datasheet** [46]. En ella se encuentra información muy importante respecto al funcionamiento del microcontrolador y, con ello, del dispositivo. Esto es porque en el **datasheet** se indica que el microcontrolador dispone de interfaz de **debug** o **JTAG** y de **serie** o **UART**. Con lo cual los seis pines descubiertos en la placa del dispositivo pueden ser la **interfaz de debug** o la **interfaz serie**.

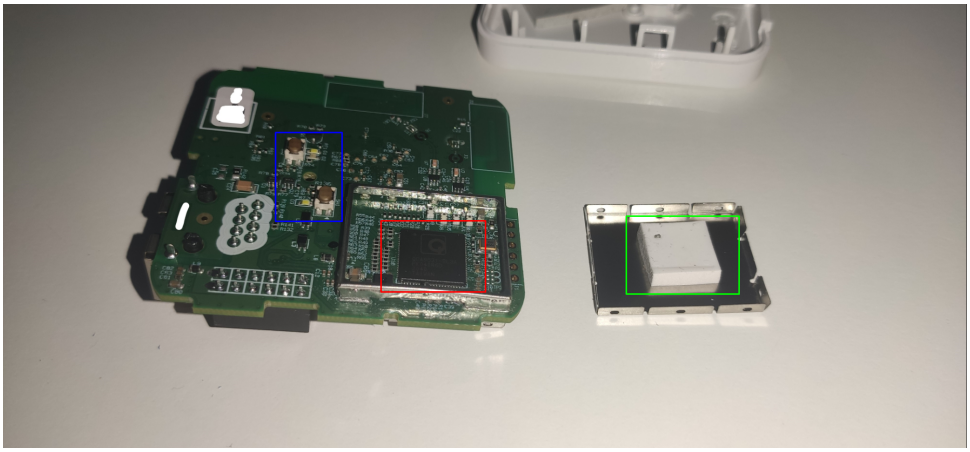

Figura 4.44: Cara delantera de la PCB del dispositivo.

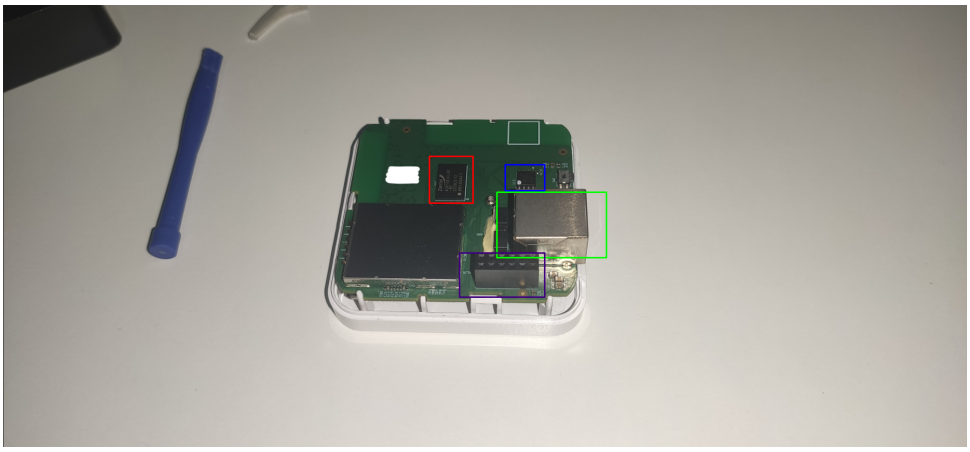

Figura 4.45: Cara trasera de la PCB del dispositivo.

Respecto a la memoria **flash**, se puede encontrar cierta información que indica la posibilidad de configurar la memoria para establecer parámetros de arranque llamadas **Boostrap Options**. Por tanto, el ataque físico constaría de vulnerar la interfaz encontrada de manera que se puedan modificar parámetros de arranque y analizar el tráfico que circule por esa interfaz.

Investigando el modelo y el manual de este dispositivo (no se hará cita del enlace debido a que revelaría el nombre de este) también se puede averiguar que los pines expuestos se utilizan para la interfaz serie donde, para este modelo en concreto: el pin 2 es salida de 12 V (voltios), el pin 3 es la toma de tierra, el pin 4 es el pin **TX** y el pin 5 es el pin **RX**. Los pines 1 y 6 están identificados pero sin información al respecto. En la Figura 4.48 se refleja esta información.

Una vez identificados los pines, se procede a intentar interactuar con la interfaz serie del dispositivo mediante la lectura/escritura en el. Pero para ello se necesita saber el **baudrate** (en español, velocidad de transmisión de datos) al que circulan los datos. Para ello, se realiza la conexión a los pines **UART** mediante un analizador lógico para analizar una muestra de transmisión de datos (por ejemplo, el arranque) y, utilizando la siguiente guía [34], determinar la velocidad de transmisión de datos que utiliza. Una vez averiguado que esta velocidad es **115.200** baudios, se puede realizar la conexión mediante un adaptador USB a interfaz serie, como se ve en la Figura 4.49, que permite tanto lectura como escritura. Esta conexión se establece con el programa **Serial Port Monitor**. Seguido de ello, se conecta de nuevo el dispositivo a la red eléctrica y se analiza y recopila toda la información que aparece en el arranque del dispositivo. La información más importante es que el sistema de arranque del dispositivo se basa en **U-Boot**.

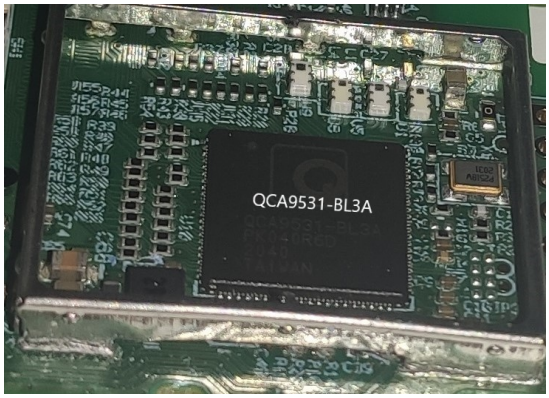

Figura 4.46: Componentes bajo la cubierta de la cara delantera de la PCB.

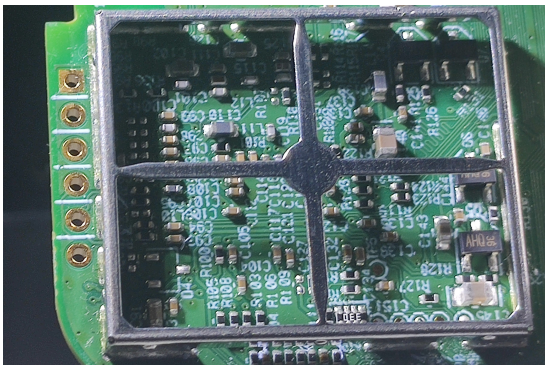

Figura 4.47: Componentes bajo la cubierta de la cara trasera de la PCB.

Pin1	R _D-	R _D+	R _D+ / F _RxD
Pin2	+12Vdc (default out) 3W)	+12Vdc 3W)	nc
Pin3	GND	GND	GND
Pin4	R _TxD	F _TxD	nc
Pin5	F _RxD	R _RxD	nc
Pin6	R _D+	R _D-	F _D- / F _TxD

Figura 4.48: Orden y clasificación de pines de la interfaz serie del dispositivo.

Una vulnerabilidad que se puede encontrar en este proceso es la de interrumpir el arranque del dispositivo y poder ejecutar comandos en la terminal del sistema de arranque. Este dispositivo no tiene protección para ello, con lo cual con introducir la tecla **intro** varias veces mientras está iniciando y antes de cargar el sistema operativo se puede interrumpir el arranque y conseguir control de la terminal **U-Boot** del dispositivo. Esta terminal ofrece una amplia variedad de comandos que se pueden ver en su página oficial [13] o escribiendo el comando **help**, ayuda en inglés, en la terminal. Algunos de esos comandos permiten modificar las variables de arranque, leer los sectores de memoria (Figura 4.50) o copiar todo el contenido de la memoria. En estas variables de arranque se encuentra la información por defecto del dispositivo, como la contraseña asociada de fábrica.

Previamente se había identificado que tenía una memoria **flash**. Esta memoria se puede desoldar y volcar su contenido. Pero no hay necesidad de ello si se puede realizar el volcado desde la interfaz serie.

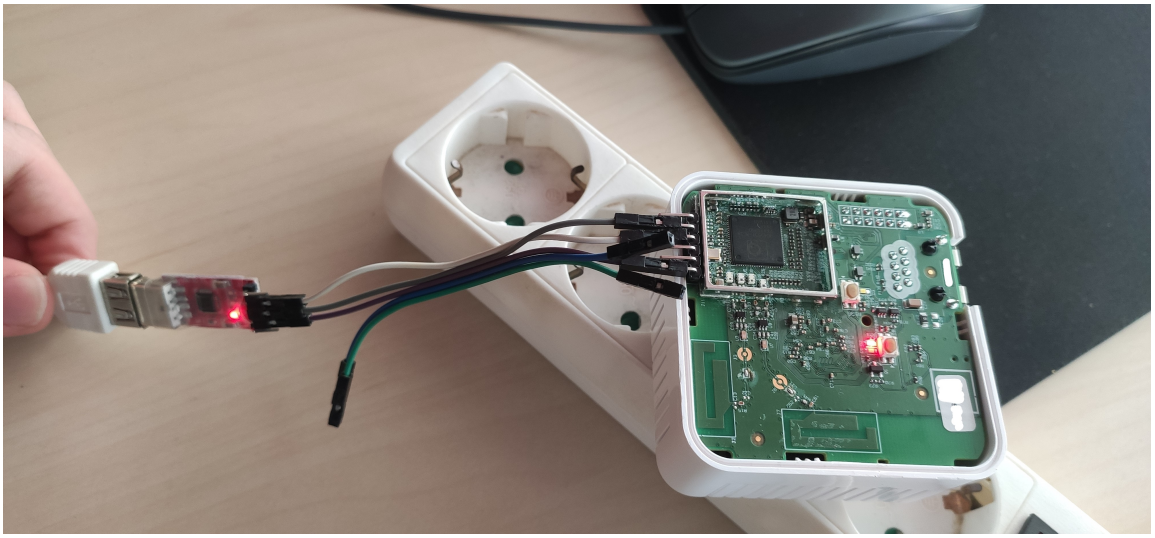

Figura 4.49: Conexión del ordenador al dispositivo mediante un adaptador USB a interfaz serie.

```
ath>  md.b 0x9f040000 100
9f040000: d1 1c f6 0e 62 6f 6f 74 63 6d 64 3d 62 6f 6f 74    ....bootcmd=boot
9f040010: 6d 20 30 78 39 66 30 37 30 30 30 30 00 62 6f 6f    m 0x9f070000.boo
9f040020: 74 64 65 6c 61 79 3d 31 00 62 61 75 64 72 61 74    tdelay=1.baudrat
9f040030: 65 3d 31 31 35 32 30 30 00 65 74 68 61 64 64 72    e=115200.ethaddr
9f040040: 3d 30 78 30 30 3a 30 78 61 61 3a 30 78 62 62 3a    =0x00:0xaa:0xbb:
9f040050: 30 78 63 63 3a 30 78 64 64 3a 30 78 65 65 00 69    0xcc:0xdd:0xee.i
9f040060: 70 61 64 64 72 3d 31 39 32 2e 31 36 38 2e 30 2e    paddr=192.168.0.
9f040070: 32 34 39 00 73 65 72 76 65 72 69 70 3d 31 39 32    249.serverip=192
9f040080: 2e 31 36 38 2e 30 2e 31 30 30 00 64 69 72 3d 00    .168.0.100.dir=.
9f040090: 6c 75 3d 74 66 74 70 20 30 78 38 30 30 36 30 30    lu=tftp 0x800600
9f0400a0: 30 30 20 24 7b 64 69 72 7d 75 2d 62 6f 6f 74 2e    00 ${dir}u-boot.
9f0400b0: 62 69 6e 26 26 65 72 61 73 65 20 30 78 39 66 30    bin&&erase 0x9f0
9f0400c0: 30 30 30 30 30 20 2b 24 66 69 6c 65 73 69 7a 65    00000 +$filesize
9f0400d0: 26 26 63 70 2e 62 20 24 66 69 6c 65 61 64 64 72    &&cp.b $fileaddr
9f0400e0: 20 30 78 39 66 30 30 30 30 30 30 20 24 66 69 6c     0x9f000000 $fil
9f0400f0: 65 73 69 7a 65 00 6c 66 3d 74 66 74 70 20 30 78    esize.lf=tftp 0x
```

Figura 4.50: Lectura de los sectores iniciales de la memoria que contiene la información de arranque del dispositivo.

4.2.6. Resumen de vulnerabilidades

Después de realizar los tres focos de ataque hacia el dispositivo, se resumen y se puntúan las vulnerabilidades encontradas cada una en su categoría. Estas puntuaciones se realizan mediante la calculadora de puntuación [40] del **Instituto Nacional de Estándares y Tecnología**, donde según la puntuación obtenida, se clasifica la severidad de las vulnerabilidades en: ninguna (0 puntos), baja (de 0.1 a 3.9 puntos), media (de 4.0 a 6.9 puntos), alta (de 7.0 a 8.9 puntos) y crítica (de 9.0 a 10.0 puntos) [39].

- **Vulnerabilidades encontradas en la auditoría remota**: En relación al ataque remoto, se ha analizado el firmware. No se encontraron vulnerabilidades pero sí se puede aportar

cierta información acerca de lo encontrado. El firmware es de descarga pública y solamente está comprimido, con lo cual cualquier usuario con acceso a la descarga del firmware podría realizar el mismo análisis. Pero además de analizar el firmware en sí, en el **ataque local** se han encontrado algunas funciones que están almacenadas en el firmware. Posiblemente estas funciones se puedan desensamblar para poder encontrar más información al respecto. Como **no representa** una **vulnerabilidad**, su **puntuación** es **0.0**.

- **Vulnerabilidades encontradas en la auditoría local**: Se han encontrado diversas vulnerabilidades en esta auditoría. La **primera** es que la administración del dispositivo está desprotegida de fábrica. Es decir, no dispone ni de usuario ni de contraseña de administrador. Por tanto, cualquier atacante que tenga acceso a la red del propio dispositivo o que conozca su dirección IP, puede vulnerar el correcto funcionamiento del dispositivo causando una **denegación de servicio** [38]. También puede obtener las credenciales del dispositivo por esta vulnerabilidad, entre otros muchos métodos descubiertos en el análisis del firmware. La **puntuación** de esta vulnerabilidad es de **8.8**, lo cual indica que esta vulnerabilidad es de riesgo **alto**. Las otras vulnerabilidades encontradas tienen relación con el tráfico de red que se realiza. Todo el tráfico de red del dispositivo se realiza a través del **puerto 80** o **HTTP**. Este protocolo es fácil de vulnerar mediante espionaje de tráfico. Por contra, el **puerto 443** o **HTTPS** sí utiliza protocolos de seguridad ante el espionaje de tráfico de red. Este protocolo es el **SSL** [14] (del inglés *Secure Sockets Layer*, Capa de *Sockets* Segura), el cual permite la encriptación y autenticación de los datos transmitidos, de manera que el envío de datos sensibles se puede realizar de manera segura. El hecho de que el tráfico circule de manera insegura es una vulnerabilidad, por tanto esta tiene una **puntuación** de **5.4**, clasificándose así en riesgo **medio**. No obstante, si se filtran datos en claro a través de este tráfico es más grave. Esta es la **otra vulnerabilidad encontrada** en este ataque, ya que se han realizado peticiones donde la contraseña de red del dispositivo ha circulado a través de la red en claro. Por tanto, esta vulnerabilidad obtiene una **puntuación** de **7.1**, la cual también se clasifica en el riesgo **alto**. Por último, otra vulnerabilidad encontrada es la de encontrar datos sensibles sin cifrar en el archivo de copia de seguridad del dispositivo. Esta vulnerabilidad recibe una **puntuación** de **8.1** y tiene un riesgo de nivel **alto** para el usuario.

- **Vulnerabilidades encontradas en la auditoría física**: Las vulnerabilidades encontradas en el ataque físico se relacionan principalmente con la interfaz serie descubierta. El hecho de que la **interfaz serie** esté **habilitada** (es decir, que exista físicamente en la placa base) en un dispositivo en su fase de producción es una **vulnerabilidad**. Además, por ello se puede interferir con el arranque del dispositivo mediante un adaptador de lectura/escritura vía **UART**. Por tanto, esta vulnerabilidad obtiene una **puntuación** de **3.5**, de manera que su riesgo es **bajo**. Se ha descubierto también que el arranque está desprotegido y que se pueden explotar los comandos de la terminal del dispositivo. Con ello pueden aparecer muchas más vulnerabilidades como consecuencia, por ejemplo, establecer una conexión remota del dispositivo. Según las métricas ofrecidas en la página citada anteriormente, para esta vulnerabilidad se obtiene una **puntuación** de **4.3** y, por tanto, un riesgo **medio**. Por último, se puede leer todo el contenido de memoria a través de esta vulnerabilidad, que contiene todos los parámetros preestablecidos del dispositivo a la hora de iniciarse, como la contraseña de red. De esta manera, esta vulnerabilidad obtiene una **puntuación** de **5.2**, con un riesgo de nivel **medio**.

En una auditoría también se debe documentar todo aquello que no se ha podido hacer. Por tanto, debido a factores como la falta de tiempo o de experiencia, no se pudo auditar el dispositivo por completo. Los puntos no auditados son los siguientes:

- Se pueden investigar todos archivos del firmware en busca de parámetros preestablecidos, como contraseñas.
- Se puede auditar de manera remota todas las conexiones del dispositivo. Esto solamente se hizo de manera local.
- Se puede intentar realizar inyecciones de código investigando las estructuras del firmware una vez desensamblado.
- Se pueden cambiar los parámetros de arranque en busca del control total del dispositivo. De esta manera se podría obtener la posibilidad de establecer una conexión SSH [51] (del inglés *Secure Shell*, Terminal Segura). Esta conexión permite conectarse a otro dispositivo de manera remota y segura. De esta manera, si se consigue esta conexión se puede investigar si hay alguna manera de conseguir control total sobre el dispositivo de manera remota.
- Se puede auditar el protocolo WPS.
- Se puede realizar un **Buffer Overflow** (en castellano, desbordamiento de memoria) [60].
- Se puede modificar el firmware y probar a actualizar el dispositivo con ese firmware.
- Se puede probar el modo a prueba de fallos del dispositivo y auditarlo.
- Se puede investigar la relación de la contraseña del usuario root de los ficheros **shadow** y **passwd** con la contraseña de administración de la página de configuración del PLC.

4.2.7. Redacción del informe final

Finalmente, se redacta el resumen ejecutivo de la auditoría. Este recopila todas las vulnerabilidades ordenadas desde las más graves (según la puntuación realizada) hasta las que menos, incluyendo los puntos meramente informativas. Una manera de recopilar estas vulnerabilidades es mediante una tabla, como la Tabla 4.2.

Código	Vulnerabilidad	Riesgo	Puntuación	Vector
VL-01	Administración insegura del dispositivo	Alto	8.8	Local
VL-02	Datos sensibles sin cifrar en el archivo de copia de seguridad	Alto	8.1	Local
VL-03	Contraseña sin cifrar en el tráfico de red	Alto	7.1	Local
VL-04	Tráfico por protocolo inseguro	Medio	5.4	Local
VF-01	Volcado de memoria a través de la interfaz de debug	Medio	5.2	Físico
VF-02	Arranque del sistema desprotegido	Medio	4.3	Físico
VF-03	Interfaz serie habilitada en producción	Bajo	3.5	Físico
INF-01	Métodos relacionados con el funcionamiento del dispositivo encontrados	Nulo	0.0	Remoto

Tabla 4.2: Resumen de vulnerabilidades encontradas en la auditoría del PLC.

Las **soluciones** a estas vulnerabilidades son las siguientes:

- Para la vulnerabilidad **VL-01**: ofrecer un usuario y una contraseña de administración de fábrica, obligando al usuario a introducir una nueva contraseña personalizada que cumpla con los estándares de seguridad de contraseñas.
- Para la vulnerabilidad **VL-02**: Cifrar todos aquellos datos sensibles que se puedan almacenar en la copia de seguridad.
- Para la vulnerabilidad **VL-03**: No transmitir datos sensibles a través de la red. En caso de transmitirlos, que sean cifrados o asegurándose que se realicen mediante protocolos de tráfico seguros.
- Para la vulnerabilidad **VL-04**: Redirigir todo el tráfico que se realice por el puerto de tráfico de red **HTTP** hacia **HTTPS** para garantizar la seguridad de este.
- Para la vulnerabilidad **VF-01**: Deshabilitar la opción de **debug** en la fase de producción.
- Para la vulnerabilidad **VF-02**: Proteger mediante contraseñas el sistema de arranque del dispositivo.
- Para la vulnerabilidad **VF-03**: Desarrollar un diseño de la placa base que no ofrezca una interfaz **serie** (o **debug**) para el dispositivo en fase de producción.
- El código **INF-01** representa la sección informativa realizada en el análisis del firmware.

De esta manera concluye la parte del auditor en la redacción del informe final de la auditoría del dispositivo PLC.

Capítulo 5

Conclusiones y trabajo futuro

Por último, en este capítulo se recopilarán las conclusiones finales del proyecto desarrollado. Además incluirá una sección que indica el trabajo futuro que se puede implementar para este.

5.1. Ámbito formativo

Como conclusión formativa de este proyecto y de esta estancia en prácticas, debo decir que he podido descubrir la gran importancia de explorar los riesgos de los dispositivos IoT. Siempre se ha hablado de que debemos protegernos en Internet tomando todas las medidas y precauciones posibles. Sin embargo, nunca imaginé que la ciberseguridad de estos dispositivos tuviera la fuerte influencia que he descubierto en este proyecto, la cual aumenta más aún con el auge de estos. También he aprendido que la ciberseguridad no es solamente criptografía, como se enfoca en la carrera, sino que es una área mucho más amplio. En el caso del itinerario de ingeniería de computadores, se aplica para los dispositivos IoT. Todo este proyecto ha servido como una nueva fuente de información de la que se puede descubrir y aprender mucho más de lo que se pueda imaginar, todo ello relacionado con el ámbito *hardware*. Por último, destacar que he conseguido mis objetivos en este proyecto, aunque me he quedado con más ganas de seguir auditando dispositivos.

5.2. Ámbito profesional

Personalmente, he aprendido mucho gracias a tener un supervisor que me ha enseñado de la mejor manera posible, ofreciéndome todo tipo de guías, ayudas, explicaciones y todo lo que necesitara para poder comprender todo correctamente. Cuando me desviaba del foco principal, él siempre me controlaba para centrarme en lo debido a lo largo de toda la estancia en prácticas. Además ha puesto a prueba mi capacidad de ser independiente en el ámbito de las auditorías y de cómo actuar por mi cuenta, siempre estando ahí por si necesitara de su ayuda. El hecho de realizar las prácticas en estado de pandemia ha dificultado la integración en la empresa, pero el contacto con el supervisor siempre ha estado presente. También considero que he contribuido en la expansión del sector de las auditorías de dispositivos IoT mediante este proyecto de fin de grado.

5.3. Ámbito personal

Este proyecto y estancia en prácticas, no solo han cambiado totalmente mis planes a futuro, sino que me han orientado hacia entender qué es a lo que realmente me quiero dedicar. Si tengo la oportunidad de dedicarme a proyectos de este estilo en mi futuro laboral, no dudaré en ningún momento en aceptarla. Como reflexión acerca de lo aprendido, considero que sería apto enseñar e informar a la gente que no todo es tan seguro como parece, que lo sencillo de configurar suele ser inseguro a escondidas. Una vez se obtiene la visión que se aprende al realizar auditorías, incluso todo parece inseguro o *hackeable*, es una visión totalmente distinta acerca de la tecnología de hoy en día, que enseña y demuestra que no todo es tan seguro como parece. Y no solamente es importante enseñar a la gente de esta inseguridad, sino formar a los futuros ingenieros que la rama de la ciberseguridad no es solamente criptografía y matemáticas, también incluye el *hacking ético*, el *pentesting* y mantener la seguridad de lo que nos rodea. La ciberseguridad engloba campos que de primeras no se ven, y cuando se descubren se hallan múltiples vías (y, en mi opinión, de las más importantes) de ella y de la ingeniería informática.

5.4. Trabajo futuro

Respecto al trabajo futuro de este proyecto, al haber desarrollado una metodología, esta se puede mejorar refinando el proceso. Se puede hacer mejorando las definiciones de los vectores de ataque, diseñar nuevos requisitos desde los cuales se abarquen más medios tecnológicos de nuevos dispositivos IoT que aparezcan en el futuro. Todo ello para hacer del proceso de auditoría desarrollado más amplio y genérico y, de esta manera, tener en cuenta la mayor cantidad de dispositivos hardware IoT posibles.

Bibliografía

[1] AatomicJC. Cómo realizar copias de seguridad de una aplicación android con **adb**. https://gist.github.com/AnatomicJC/e773dd55ae60ab0b2d6dd2351eb977c1. [Consulta: 26 de Julio de 2022].

[2] ADH Technology. Data sheet del módulo gt-511c2. https://www.yumpu.com/en/document/read/23268061/2-protocol-packet-structure. [Consulta: 26 de Julio de 2022].

[3] Blog Alberto, escritor anónimo. Riesgos de los puertos de red abirtos. http://alberto16482.blogspot.com/2016/04/riesgos-potenciales-en-los-servicios-de.html. [Consulta: 26 de Julio de 2022].

[4] bluetooth.com. Qué es ble. https://www.bluetooth.com/learn-about-bluetooth/tech-overview/. [Consulta: 26 de Julio de 2022].

[5] Boletín Oficial del Estado. Ley 42/2006, de 28 de diciembre, de presupuestos generales del estado para el año 2007 del boletín oficial del estado. https://www.boe.es/eli/es/l/2006/12/28/42/dof/spa/pdf. [Consulta: 26 de Julio de 2022].

[6] calcmaps.com. Calculadora de áreas basada en medición a través de bing maps. https://www.calcmaps.com/es/map-area/reeg3o/. [Consulta: 26 de Julio de 2022].

[7] circuitbasics.com. Qué es uart. https://www.circuitbasics.com/basics-uart-communication/. [Consulta: 26 de Julio de 2022].

[8] codeshare.com. Librería frida codeshare. https://codeshare.frida.re/browse. [Consulta: 26 de Julio de 2022].

[9] consultingpro.laboralkutxa.com. Cómo calcular los costes de contratación; primera fuente. https://consultingpro.laboralkutxa.com/articulos/cuanto-cuesta-contratar-a-un-trabajador/. [Consulta: 26 de Julio de 2022].

[10] cuatroochenta.com. Página oficial de soluciones cuatroochenta s.l.. https://cuatroochenta.com. [Consulta: 26 de Julio de 2022].

[11] cyberciti.biz. Cómo entender el fichero **passwd**. https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/. [Consulta: 26 de Julio de 2022].

[12] cyberciti.biz. Cómo entender el fichero **shadow**. https://www.cyberciti.biz/faq/understanding-etcshadow-file/. [Consulta: 26 de Julio de 2022].

[13] Digi International. Manual de usuario de la terminal u-boot. https://hub.digi.com/dp/path=/support/asset/u-boot-reference-manual/. [Consulta: 26 de Julio de 2022].

[14] Digicert. Qué es el protocolo ssl/tls. https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https. [Consulta: 26 de Julio de 2022].

[15] Diversos autores del repositorio en GitHub. Página de github de la herramienta **libnfc**. https://github.com/nfc-tools/libnfc. [Consulta: 26 de Julio de 2022].

[16] Diversos autores del repositorio en GitHub. Página de github de la herramienta **mfcuk**. https://github.com/nfc-tools/mfcuk. [Consulta: 26 de Julio de 2022].

[17] Diversos autores del repositorio en GitHub. Página de github de la herramienta **mfoc**. https://github.com/nfc-tools/mfoc. [Consulta: 26 de Julio de 2022].

[18] Diversos usuarios contribuyentes de GitHub. Librería milazycracker. https://github.com/nfc-tools/miLazyCracker. [Consulta: 26 de Julio de 2022].

[19] espaitec.uji.es. Calculadora de costes mensuales del edificio espaitec 2 de la uji. https://espaitec.uji.es/calculadora/. [Consulta: 26 de Julio de 2022].

[20] espaitec.uji.es. Infraestructuras del edificio espaitec 2 de la uji. https://espaitec.uji.es/infraestructuras/. [Consulta: 26 de Julio de 2022].

[21] exploit-db.com. Como realizar un análisis de firmware: Tercera parte. https://www.exploit-db.com/docs/48566. [Consulta: 26 de Julio de 2022].

[22] fortinet.com. Qué es tcp. https://www.fortinet.com/resources/cyberglossary/tcp-ip. [Consulta: 26 de Julio de 2022].

[23] fortinet.com. Qué es udp. https://www.fortinet.com/resources/cyberglossary/user-datagram-protocol-udp. [Consulta: 26 de Julio de 2022].

[24] fortinet.com. Qué es un proxy. https://www.fortinet.com/resources/cyberglossary/proxy-server. [Consulta: 26 de Julio de 2022].

[25] geeksforgeeks.org. Definición de ram (random access memory). https://www.geeksforgeeks.org/random-access-memory-ram/. [Consulta: 26 de Julio de 2022].

[26] geeksforgeeks.org. Definición de rom (read-only memory). https://www.geeksforgeeks.org/read-only-memory-rom/. [Consulta: 26 de Julio de 2022].

[27] glassdor.es. Salario de analista de ciberseguridad junior. https://www.glassdoor.es/Sueldos/analista-de-ciberseguridad-junior-sueldo-SRCH_KO0,33.htm. [Consulta: 26 de Julio de 2022].

[28] Google Developers. Qué es una **cookie** de identificación y para qué sirve. https://developers.google.com/tag-platform/devguides/cookies. [Consulta: 26 de Julio de 2022].

[29] Grupo de hacking ético y de ciberseguridad, SensePost. Librería objection. https://github.com/sensepost/objection/wiki/Using-objection. [Consulta: 26 de Julio de 2022].

[30] Hangzhou Grow Technology. Manual de usuario del módulo r307. https://www.openhacks.com/uploadsproductos/r307_fingerprint_module_user_manual.pdf. [Consulta: 26 de Julio de 2022].

[31] Hangzhou Grow Technology. Manual de usuario del módulo r502. https://www.dropbox.com/sh/epucei8lmoz7xpp/AAAmon04b1DiSOeh1q4nAhzAa?dl=0&preview=R502+fingerprint+module+user+manual-V1.2.pdf. [Consulta: 26 de Julio de 2022].

[32] infoautonomos.com. Cómo calcular los costes de contratación; segunda fuente. https://www.infoautonomos.com/blog/cuanto-cuesta-contratar-un-trabajador/. [Consulta: 26 de Julio de 2022].

[33] kb.iweb.com. Qué es mdns. https://kb.iweb.com/hc/es/articles/360005117952-Guia-para-resolver-problemas-deseguridad-de-Multicast-DNS-mDNS. [Consulta: 26 de Julio de 2022].

[34] kumari.net. Cómo averiguar la velocidad de transmisión de datos. https://www.kumari.net/index.php/random/37-determing-unknown-baud-rate. [Consulta: 26 de Julio de 2022].

[35] medium.com. Como realizar un análisis de firmware: Segunda parte. https://medium.com/@attify/firmware-analysis-for-iot-devices-fb8df961c19d. [Consulta: 26 de Julio de 2022].

[36] medium.com. Cómo preparar el entorno de auditoría del tráfico web de aplicaciones móviles. https://medium.com/swlh/android-mobile-penetration-testing-lab-dfb8ceb4efbd. [Consulta: 26 de Julio de 2022].

[37] movistar.es. Plan de internet utilizado para el proyecto. https://www.movistar.es/empresas/para-tu-oficina/conectividad-internet/adsl-empresas/. [Consulta: 26 de Julio de 2022].

[38] National Cyber Security Centre. Qué es el ataque de denegación de servicio (dos). https://www.ncsc.gov.uk/collection/denial-service-dos-guidance-collection. [Consulta: 26 de Julio de 2022].

[39] National Institute Of Standards And Technology. Clasificación de las vulnerabilidades según la puntuación obtenida. https://nvd.nist.gov/vuln-metrics/cvss. [Consulta: 26 de Julio de 2022].

[40] National Institute Of Standards And Technology. Cálculo de la puntuación de las vulnerabilidades. https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator. [Consulta: 26 de Julio de 2022].

[41] OPM Integral. Qué es la metodología predictiva. https://opmintegral.com/gestion-de-proyectos/metodologias-agiles-vs-tradicionales/. [Consulta: 26 de Julio de 2022].

[42] pentesters.com. Como realizar un análisis de firmware: Primera parte. https://www.pentestpartners.com/security-blog/how-to-do-firmware-analysis-tools-tips-and-tricks/. [Consulta: 26 de Julio de 2022].

[43] poftut.com. Cómo escanear puertos de red. https://www.poftut.com/how-to-scan-all-tcp-and-udp-ports-with-nmap/. [Consulta: 26 de Julio de 2022].

[44] portswigger.net. Cómo configurar el certificado de burpsuite. https://portswigger.net/support/installing-burp-suites-ca-certificate-in-an-android-device. [Consulta: 26 de Julio de 2022].

[45] portswigger.net. Cómo configurar el móvil para hacerlo funcionar con burpsuite. https://portswigger.net/support/configuring-an-android-device-to-work-with-burp. [Consulta: 26 de Julio de 2022].

[46] Qualcom Atheros. Datasheet del microcontrolador. https://datasheet.lcsc.com/szlcsc/Qualcomm-QCA9531-BL3A_C135781.pdf. [Consulta: 26 de Julio de 2022].

[47] Saleae. Página oficial de saleae. https://www.saleae.com/es/. [Consulta: 26 de Julio de 2022].

[48] Slawomir Jasek. Guía práctica para hackear rfid y nfc. https://smartlockpicking.com/slides/Confidence_A_2018_Practical_Guide_To_Hacking_RFID_NFC.pdf. [Consulta: 26 de Julio de 2022].

[49] sofistic.com. Página oficial de sofistic. https://www.sofistic.com. [Consulta: 26 de Julio de 2022].

[50] SparkFun Electronics. Sensor de huella de sparkfun electronics. https://forum.sparkfun.com/viewtopic.php?t=12432&start=105. [Consulta: 26 de Julio de 2022].

[51] ssh.com. Página oficial de ssh. https://www.ssh.com/academy/ssh/protocol. [Consulta: 26 de Julio de 2022].

[52] ssl.com. Qué es https. https://www.ssl.com/faqs/what-is-https/. [Consulta: 26 de Julio de 2022].

[53] techopedia.com. Qué es nfc. https://www.techopedia.com/definition/27583/near-field-communication-nfc. [Consulta: 26 de Julio de 2022].

[54] techopedia.com. Qué son los puertos de red. https://www.techopedia.com/definition/24717/network-port. [Consulta: 26 de Julio de 2022].

[55] techtarget.com. Qué es rfid. https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-identification. [Consulta: 26 de Julio de 2022].

[56] tulane.edu. Cálculo de tiempo para realizar ataques de fuerza bruta sobre contraseñas. https://tmedweb.tulane.edu/content_open/bfcalc.php. [Consulta: 26 de Julio de 2022].

[57] Usuario de GitHub ikarus23. Repositorio de mifare classic tool. https://github.com/ikarus23/MifareClassicTool. [Consulta: 26 de Julio de 2022].

[58] w3schools.com. Qué es http. https://www.w3schools.com/whatis/whatis_http.asp. [Consulta: 26 de Julio de 2022].

[59] wakdev. Página oficial de descarga de la aplicación nfc tools. https://play.google.com/store/apps/details?id=com.wakdev.wdnfc. [Consulta: 26 de Julio de 2022].

[60] welivesecurity.com. Qué es un buffer overflow. https://www.welivesecurity.com/la-es/2014/11/05/como-funcionan-buffer-overflow/. [Consulta: 26 de Julio de 2022].

[61] ZhianTec. Manual de usuario del módulo zfm-70. https://www.velleman.eu/downloads/29/infosheets/vma329_datasheet.pdf. [Consulta: 26 de Julio de 2022].

Anexo A

BLE (Bluetooth Low Energy)

attify.com. Guía práctica para hackear BLE. https://blog.attify.com/the-practical-guide-to-hacking-bluetooth-low-energy/. [Consulta: 26 de Julio de 2022].

Arun M. Magesh. Como realizar ingeniería inversa a una pulsera inteligente con BLE. https://medium.com/@arunmag/my-journey-towards-reverse-engineering-a-smart-band-bluetooth-le-re-d1dea00e4de2. [Consulta: 26 de Julio de 2022].

Arun M. Magesh. Cómo realizar ingeniería inversa (y vulnerar) un masajeador inteligente. https://medium.com/@arunmag/how-i-reverse-engineered-and-exploited-a-smart-massager-ee7c9f21bf33. [Consulta: 26 de Julio de 2022].

Anexo B

RFID (Radio-frequency identification)

Usuario octosavvi de GitHub. Configurar ESPKey para ESP. https://github.com/octosavvi/ESPKey. [Consulta: 26 de Julio de 2022].

Diversos usuarios de GitHub. Configurar RFID para ESP. https://github.com/rfidtool/ESP-RFID-Tool. [Consulta: 26 de Julio de 2022].

Anexo C

Aplicaciones móviles

Alexey Alter-Pesotskiy. Obtener archivo ".ipa" para las aplicaciones iOS. https://medium.com/testableapple/how-to-download-ipa-from-app-store-43e04b3d0332. [Consulta: 26 de Julio de 2022].

Diversos usuarios de GitHub. Realizar debug con OpenOCD. https://github.com/openocd-org/openocd. [Consulta: 26 de Julio de 2022].