UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Auditor Wifi desde Raspberry Pi controlado por dispositivo Android

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Pablo Adrián Moreno Sierra

Tutor: Carlos Tavares Calafate

Curso: 2014/2015

Resumen

En este proyecto se propone adaptar una Raspberry Pi 2 con el sistema operativo Kali Linux para poder utilizar las aplicaciones de auditoria Wifi en movilidad. El cliente es un dispositivo Android, desde el cual se gestiona el servidor integrado en la Raspberry Pi 2 mediante botones y listas para la selección de opciones, enviando así comandos Bash para ejecutar en el servidor. La comunicación cliente-servidor se realiza mediante Bluetooth, y tanto en el cliente como en el servidor se basan mayormente en el lenguaje de programación Java.

Palabras clave: Android, Wifi, Bluetooth, Raspberry Pi, Kali Linux

Summary

This Project proposes adapting a Raspberry Pi 2 with the Kali Linux operating system to use Wifi audit tools on mobility. The client is an Android device that manages the server integrated in the Raspberry Pi 2 through buttons and lists for the choice of options, sending to the server Bash commands for execution. The client-server communication relies on Bluetooth, and the programming language used on both client and server is Java.

Key words: Android, Wifi, Bluetooth, Raspberry Pi, Kali Linux

Tabla de contenido

Índice de figuras

1. Introducción

Desde hace mucho tiempo la seguridad en las TIC es algo muy importante ya que no es agradable que cualquier persona, y menos un malhechor, se haga con datos personales, bancarios o conversaciones.

En cuanto a la comunicación Wifi, el estándar original publicado en 1999 utiliza WEP (Wired Equivalent Privacy) para realizar autenticación y cifrado de datos. Dada la inseguridad de este sistema de cifrado, en 2003 se ha publicado un método alternativo para cifrar la comunicación Wifi denominado WPA (Wifi Protected Access), y finalmente en 2004 apareció el WPA2 (Wifi Protected Access 2), para que de manera definitiva el cifrado WEP quedara revocado y sin uso. A pesar de ello, a día de hoy, aun es posible encontrar APs (Access Points) utilizando WEP.

En 2007 se ha propuesto la solución conocida como WPS (Wifi Protected Setup), mediante la cual, un usuario se puede conectar a su AP sin introducir la clave correspondiente mediante diferentes técnicas:

- PIN: Se produce un intercambio de credenciales al introducir el PIN asignado.
- PBC: Se produce un intercambio de credenciales al presionar un botón en el AP.
- NFC: Se produce un intercambio de credenciales al producirse una comunicación NFC.
- USB: Se produce un intercambio de credenciales mediante un dispositivo USB.

En 2011 se descubrió una vulnerabilidad en WP, mediante la cual un atacante puede tratar de recuperar el PIN WPS en pocas horas, con él la clave pre-compartida de este AP con cifrado WPA/WPA2, usando ataques de fuerza bruta.

1.1. Motivación

Desde hace bastantes años el tema de la seguridad informática me ha interesado. De hecho, desde el instituto intentaba penetrar en redes Wifi, seguía blogs de seguridad, etc. Casi siempre el tema de romper la seguridad se ha llevado a cabo desde Linux, por eso, cuando hace no mucho descubrí la Raspberry Pi, pensé que podía ser buena idea utilizarla como Auditor, evitando así tener que llevar siempre un ordenador encima o no poder usarlo mientras estoy auditando dado el gran tiempo que conllevan algunos ataques. Además, el reducido tamaño de este computador ofrece ventajas en términos de movilidad, por lo que así comienza este proyecto.

1.2. Objetivos

El objetivo principal de este proyecto consiste en el diseño e implementación de una aplicación para dispositivos móviles *Android*, que permita manejar remotamente una Raspberry Pi para así atacar y comprobar la seguridad en las redes Wifi y en las transmisiones a través de Internet mediante ataques del tipo *Man in the middle*.

- En cuanto a la seguridad en las redes Wifi, se trata de hacer un ataque para intentar obtener su clave pre-compartida y poder conectarse a ella.
- Respecto a las transmisiones a través de Internet mediante *Man in the middle*, se trataría de utilizar *sniffers* y otras herramientas para intentar obtener los posibles datos importantes que puedan circular, como por ejemplo pueden ser las claves de acceso a diversos sitios web.

1.3. Estructura de la memoria

La presente memoria está organizada en un conjunto de secciones o capítulos, los cuales contienen información acerca de cada uno de los aspectos concretos que componen el proyecto. A continuación, se describe de forma breve de qué trata cada uno de ellos:

• Capítulo 1 - Introducción: Esta sección pretende situar al lector en el contexto en el que se desarrolla el proyecto, exponiendo la idea general en torno a la que gira, los objetivos que persigue, y las principales motivaciones que han impulsado su realización.

• Capítulo 2 - Estudio de soluciones existentes para ataques a redes Wifi: En este apartado se presenta de manera general las soluciones existentes actualmente para realizar este tipo de ataques y se realiza un breve repaso sobre otros trabajos desarrollados en el mismo ámbito.

• Capítulo 3 - Propuesta de sistema embebido para ataques a redes wifi: Este apartado describe con precisión la arquitectura del sistema y los distintos elementos que la componen.

• Capítulo 4 - Detalles sobre distribución para Raspberry Pi: En esta sección se muestran los detalles de implementación de los principales elementos que componen el sistema servidor.

• Capítulo 5 - Herramienta de gestión en Android: En esta sección se muestran los detalles de implementación de los principales elementos que componen el sistema cliente.

- Capítulo 6 - Pruebas: En este epígrafe se describen las pruebas realizadas para validar el sistema, el consumo energético y los resultados obtenidos.

- Capítulo 7 - Conclusiones: En este capítulo se analizan los resultados logrados frente a los objetivos marcados al comienzo, y se establecen las futuras líneas de trabajo.

- Capítulo 8 - Bibliografía: En la última sección se enumeran los recursos bibliográficos a los que se ha recurrido para la elaboración del trabajo.

2. Estudio de soluciones existentes para ataques a redes Wifi y Man in the middle

2.1. Linux

Para entornos basados en Linux están disponibles la gran mayoría de herramientas para auditoría de seguridad informática, e incluso existen diversos sistemas operativos diseñados específicamente para comprobar la seguridad en las TIC en todos sus aspectos. Entre estos podemos encontrar *Kali Linux* (Antiguo *Backtrack*), que quizás sea el más conocido en este ámbito, y el cual incluye una gran cantidad de herramientas para la auditoría de seguridad, además de una gran compatibilidad con diferentes procesadores (Intel x86 y x64, ARM…). Por ello me he centrado en este sistema operativo a la hora de desarrollar el proyecto.

A la hora de hacer un ataque del tipo *Man in the middle* en Linux tenemos una gran variedad de herramientas, entre las cuales destacan "Ettercap", que a través de otras herramientas, es capaz de redireccionar el tráfico por tu computadora, inyectar paquetes, hacer de *sniffer*, e incluso es capaz de "descifrar" el contenido cifrado en SSL, como puede ser el caso de *https*, mediante la herramienta "sslstrip". Realmente lo que hace no es descifrar, sino que trata de engañar al servidor para que la conexión no sea cifrada, es decir, para que utilice *http* en lugar de *https*.

Si nos centramos en la auditoría Wifi los sistemas operativos más conocidos son *WifiWay*, el cual ya quedó obsoleto, y *WifiSlax*, una distribución *Slackware* con una gran cantidad de herramientas de auditoría Wifi y scripts para facilitar y acelerar su utilización.

2.2. Windows

Este sistema operativo, a pesar de ser dónde más vulnerabilidades encontramos debido al gran número de usuarios que tiene, no posee muchas herramientas de auditoría, aunque las herramientas más famosas de Linux sí tienen su versión para Windows, como por ejemplo la suite *aircrack-ng*. Dicha herramienta es quizás la más conocida en cuanto a auditorías Wifi como ya detallaré más adelante. Esta herramienta, a pesar de tener su versión para Windows, no es capaz de realizar las mismas funciones ya que no puede poner la tarjeta de red inalámbrica en modo monitor para así lanzar ciertos tipos de ataques.

Por otro lado, también es posible encontrar alguna herramienta más desarrollada en este sistema operativo que en Linux, como puede ser "Intercepter-ng". Dicha herramienta se puede utilizar en ataques del tipo *Man in the middle* para capturar y analizar el tráfico, y así obtener de una manera muy rápida y sencilla claves, cookies o sesiones. Además,

puede hacer pasar una máquina por un router, redirigiendo hacia ella todo el tráfico, lo que permite analizarlo.

2.3. Android

Para Android, aparentemente se puede encontrar una gran cantidad de aplicaciones capaces de obtener datos personales de una víctima, leer sus SMS o sus *Whatsapp*, e incluso hackear redes Wifi. No obstante, si analizamos dichos programas, se puede comprobar como muchos de ellos eran *malware* o simplemente no hacían nada. Concretamente, en las apps para hackear redes Wifi, he podido observar como bastantes de éstas, simulaban un ataque utilizando comandos reales y obtenían una supuesta contraseña, obviamente al azar. También se puede encontrar una gran cantidad de apps que te dicen la contraseña por defecto de ciertos APs, como por ejemplo, los *WLANXX*. En última instancia hay muy pocas apps que realmente funcionan, e incluso así lo hacen a muy pequeña escala. Estas apps que funcionan, en primer lugar necesitan permisos de *root*, por lo que no siempre se van a poder utilizar, y una vez se han concedido estos permisos, instalan una serie de librerías para poder utilizar la tarjeta de red del móvil/Tablet en modo monitor y así poder atacar. En este caso la app "WIFI WPS WPA TESTER (ROOT)" es capaz de detectar APs con el protocolo WPS activo, y prueba unos pocos PINs por defecto, siendo por lo tanto capaz de obtener la contraseña pre-compartida en un AP con cifrado WPA/WPA2.

En Android también disponemos de una herramienta muy buena para hacer un ataque del tipo *Man in the middle* en una red Wifi. Se trata de la app "Intercepter-ng", y se trata de la misma aplicación que tenemos para Linux (muy poco desarrollada) y para Windows. Al igual que en Windows, posee una interfaz bastante entendible y fácil de usar, con la cual es posible hacer *sniffing* de paquetes de la red Wifi a la cual estás conectado para obtener contraseñas, cookies, sesiones, etc.

3. Propuesta de sistema embebido para ataques a redes Wifi

3.1. Detalles hardware y sistema operativo servidor

En este caso se dispone de una *Raspberry Pi 2 Model B* que tiene las especificaciones siguientes:

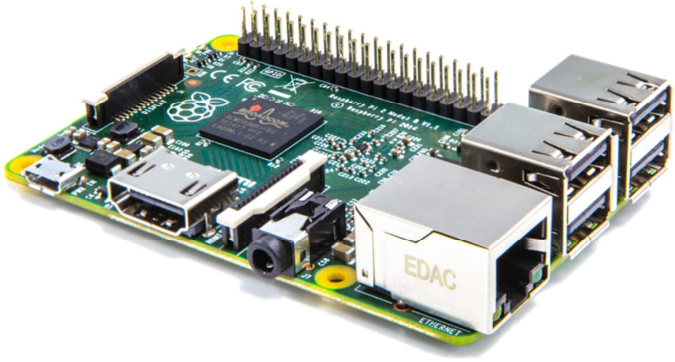

Ilustración 1: Raspberry Pi 2 model B

- Procesador ARM Cortex-A7 quad-core a 900 MHz
- Procesador gráfico VideoCore IV 3D
- 1 GB de memoria RAM
- 4 puertos USB
- 40 pines GPIO
- 1 puerto HDMI
- 1 puerto Ethernet
- 1 puerto de audio Jack 3.5 mm
- Interfaz para cámara (CSI)
- Interfaz Display (DSI)
- Ranura para tarjeta MicroSD

Además, se utilizan los siguientes periféricos:

- Adaptador Wifi por USB de 150 Mbps (Ralink RT5370) con antena externa intercambiable
- Adaptador Bluetooth v4.0 por USB
- MicroSD Samsung Pro Clase 10 de 32 Gb con 70 Mbps de velocidad de lectura
- Sistema operativo: Kali Linux armel 1.1.0a

3.2. Detalles hardware y sistema operativo cliente

Como cliente, se propone un teléfono móvil Android. Para el desarrollo de este proyecto se ha elegido el terminal *Sony Xperia Z3* con las siguientes características:

Ilustración 2: Sony Xperia Z3 Negro

- Procesador Qualcomm Snapdragon de cuatro núcleos a 2,5 GHz
- 3 GB de memoria RAM
- Pantalla de 5.2'' con una resolución de 1080x1920
- Bluetooth v4.0
- Android 5.0.2 original sin root

4. Detalles de configuración de la Raspberry Pi

4.1. Instalación y configuración del sistema operativo de la Raspberry Pi

Como ya he comentado anteriormente, se va a utilizar el sistema operativo *kali Linux*, ya que es posiblemente el sistema operativo con herramientas de seguridad más completo que se puede encontrar y además es gratuito, y en este caso aprovechamos también que posee versiones para ARM.

Dado que cuando se comenzó el proyecto acababa de salir la *Raspberry Pi 2 Model B*, no había una versión de este sistema operativo para esta computadora, lo que ocasionó problemas adicionales. Finalmente se ha optado por descargar la versión para la *Raspberry Pi 1* e instalarla en la SD. Para hacer esto, previamente hay que instalar el sistema operativo *Raspbian* puesto que, dada la incompatibilidad de la versión para la *Raspberry Pi 1* y *2*, hay que retocar la partición de arranque de la SD. En primer lugar se instala la versión *Raspbian* y se guarda la partición de arranque, y posteriormente se instala el sistema operativo *Kali Linux*.

Las instalaciones se pueden hacer de dos maneras:

1. Desde Linux: Con el comando "dd if=kali-1.0.9-rpi.img of=/dev/sdb bs=512k", siendo */dev/sdb* la partición correspondiente a la SD y *kali-1.0.9-rpi.img* la imagen del sistema operativo.
2. Desde Windows: Con el programa "Win32DiskImager", seleccionando la imagen de *Kali Linux* y la unidad correspondiente a la SD.

Cuando esto esté completado, se copian los ficheros de la partición de arranque de *Raspbian* que nos hemos guardado y se pegan en la misma partición pero con el *Kali Linux* instalado. Con esto ya tenemos un sistema operativo perfectamente funcional.

Una vez se ha instalado correctamente el sistema operativo es importante conectarlo a Internet para actualizarlo, poniendo al día drivers, kernel y por supuesto, las herramientas de las que dispone. Es también importante instalar los paquetes necesarios para tener todas las herramientas de auditoría Wifi necesarias. En este caso, al menos se necesita el paquete, *kali-linux-wireless*, el cual se puede descargar desde https://www.kali.org/news/kali-linux-metapackages/, y se puede utilizar el que más convenga. Para instalarlo solamente hay que poner el siguiente comando: "apt-get update kali-linux-wireless".

Cuando se haya hecho todo lo anterior se puede comenzar instalando las librerías para la gestión Bluetooth desde Java (Bluecove) puesto que va a ser este el lenguaje en el que se va a ejecutar el programa servidor.

En este punto encontramos otro problema: no hay versión de *Bluecove* para procesador ARM, por lo que procede compilarla. Para ello seguimos los pasos de este enlace:

http://stackoverflow.com/questions/23142071/native-library-bluecove-arm-not-available.

4.2. Programación y funcionamiento de la Raspberry Pi

Como ya se ha comentado, el programa servidor está hecho en Java, así que será necesario compilarlo si no lo estaba previamente. Para ello suponiendo que todos los ficheros "*.java" y librerías están en la carpeta "/root/" se compilará con el comando "javac -classpath .:bluecove-2.1.0.jar:bluecove-gpl-2.1.0.jar: gcm-server.jar:json-simple-1.1.1.jar *.java". También es posible utilizar un script para hacerlo más rápido.

A partir de ese momento ya se puede proceder a ejecutar el servidor, y para ello podemos utilizar la línea de comandos "java -cp 'bluecove-gpl-2.1.0.jar:bluecove-2.1.0.jar:gcm-server.jar:json-simple-1.1.1.jar:.' serverAuditorPI". Como en el caso anterior también es posible hacerlo desde un script.

Una vez iniciado el programa, el servidor espera una conexión por Bluetooth por parte de un dispositivo emparejado.

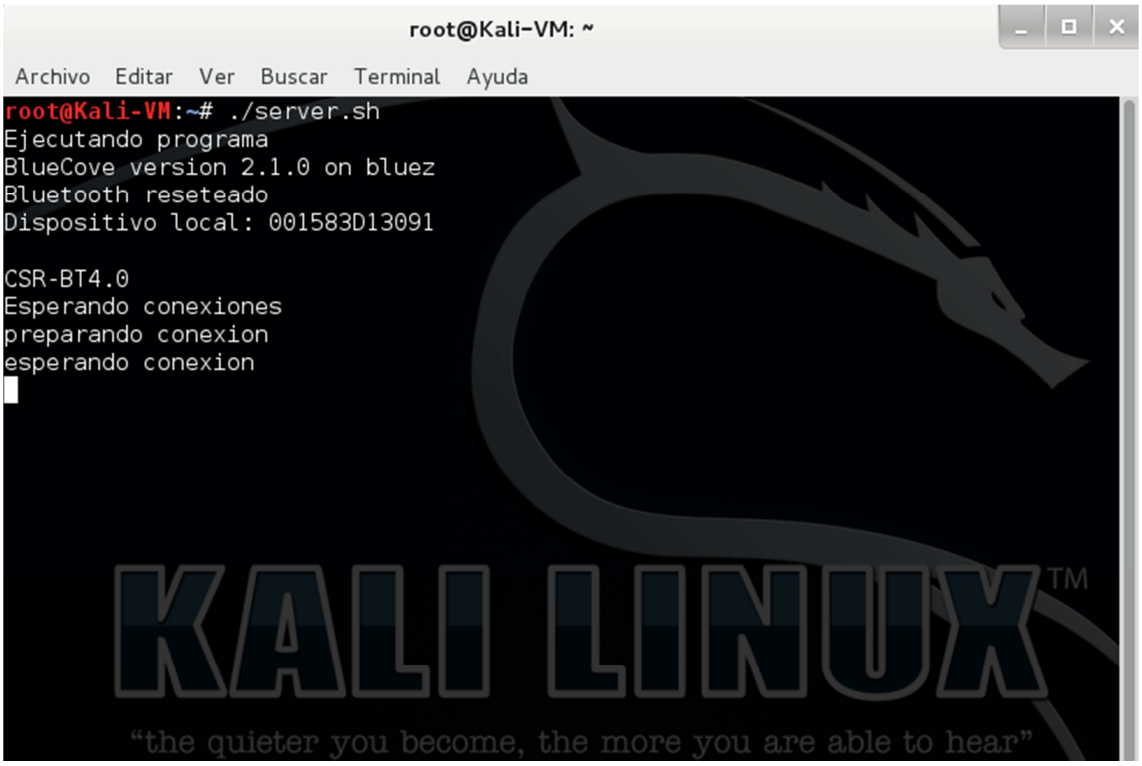

Ilustración 3: Servidor esperando conexión.

Una vez un dispositivo ha establecido conexión con la Raspberry Pi mediante Bluetooth, ambas partes se tienen que saludar y responder para verificar que ambos dispositivos conectados son los correctos y que están sincronizados en el punto de ejecución correcto.

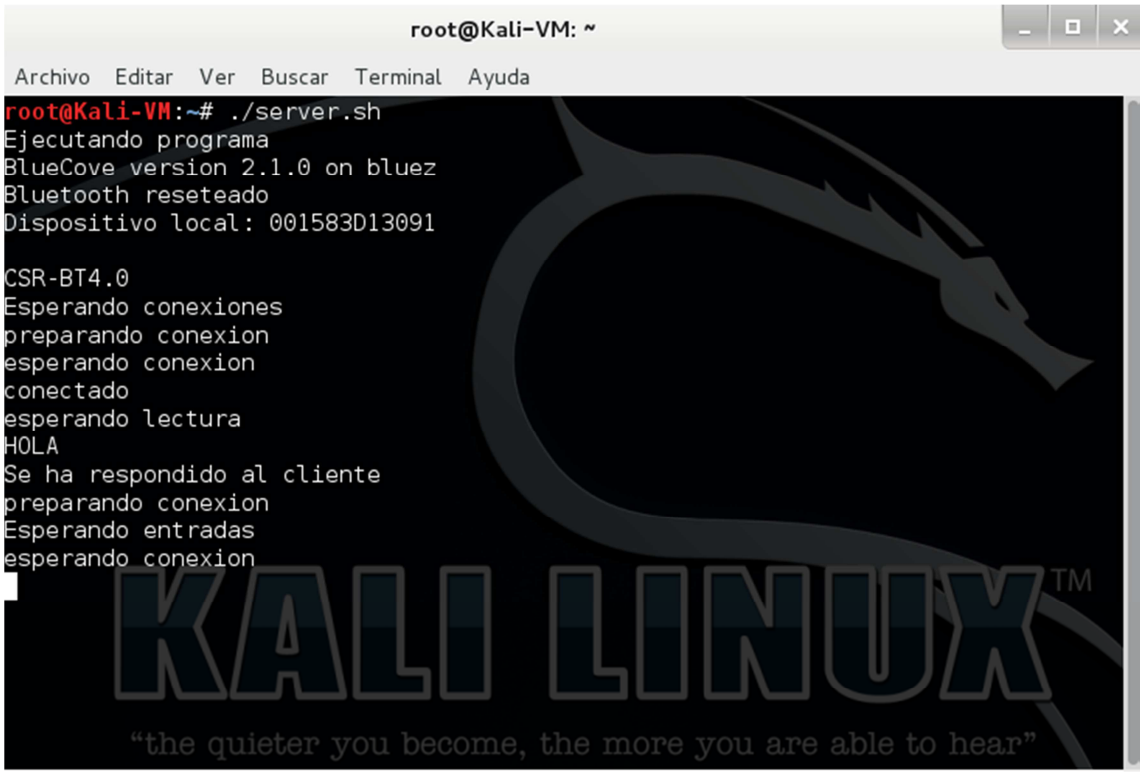

Ilustración 4: Servidor conectado.

Después de que se produzca el saludo, el servidor envía una serie de parámetros al cliente, indicando el estado actual del servidor, es decir, si se está haciendo un escaneo de redes Wifi, un ataque y de qué tipo, si ya se había puesto la tarjeta de red inalámbrica en modo monitor, e incluso si ya se ha obtenido una clave, y en tal caso la transmite al cliente.

Tras haber concluido el periodo de sincronización del estado del servidor, se crea un hilo de ejecución nuevo. En este hilo de ejecución está a la escucha del canal Bluetooth abierto, y se analizan las entradas que se producen separándolas en cuatro posibilidades según su prefijo.

Opciones:

1. Comando bash: Si el prefijo enviado coincide con el predefinido para ejecutar un comando bash, este se ejecutará, lo cual se hará mediante la siguiente línea en lenguaje Java:
   "Runtime.getRuntime().exec(new String[] {"/bin/bash", "-c",comando});".
   Al ejecutar un comando, se crea un nuevo hilo de ejecución, el cual devuelve el resultado del comando. Además se analizan los comandos enviados y sus resultados para establecer el estado actual del servidor, es decir, si está el modo monitor activado, si está escaneando, atacando, o si se ha completado un ataque.

2. Cancelar comando bash: Si la entrada de Bluetooth tiene el prefijo correspondiente al predefinido para cancelar un comando, se enviará una señal de cancelación del último comando enviado y destruyendo, por lo tanto, el hilo que había para esa ejecución. En caso de no haber un comando en ejecución no se hará nada. Además también se actualizará el estado actual del servidor.

3. Cerrar conexión: Si el prefijo en la entrada corresponde al de cerrar la conexión, la conexión por Bluetooth se cerrará, cerrando por lo tanto los hilos de escucha, y se resetea el gestor de conexiones de Bluetooth para evitar posibles fallos al volver a conectar.

4. Otros: En cualquier caso no contemplado anteriormente se ignorará dicho contenido para evitar problemas en el caso de mensajes enviados desde dispositivos o de maneras incorrectas.

Finalmente, en caso de estar ejecutando un comando bash para atacar una red Wifi y obtener su contraseña. Para evitar estar conectado mediante Bluetooth con el cliente durante todo el ataque, que como se ha mencionado anteriormente, puede durar varias horas, el servidor es capaz de enviar notificaciones *push* al cliente mediante una ID única para dicho cliente a través la API de Google.

Cuando el ataque a la red Wifi ha tenido éxito, la *Raspberry Pi* procede a conectarse a Internet mediante esa red obtenida para así verificar su correcta obtención y conexión. Tras esta verificación, el servidor avisa al cliente mediante dicha notificación de que ha obtenido la clave de la red Wifi, y la transmite a través de Internet sin necesidad de conectarse por Bluetooth. Se asume que el cliente tiene acceso a Internet por alguna otra vía alternativa (ej. Conexión 4G).

5. Herramienta de gestión en Android

La herramienta para la gestión de la *Raspberry Pi* es una *app* para Android desarrollada en *eclipse* mediante el plugin *ADT* (Android Development Kit) para la integración de Android en *eclipse* y el *Android SDK*.

Esta *app* llamada *Auditor PI* gráficamente utiliza el método *swipe* para la gestión de pantallas, es decir, que arrastrando el dedo hacia izquierda o derecha en la pantalla puedes cambiar entre las diversas pantallas y funcionalidades en este caso. Cabe destacar que ciertos aspectos de esta interfaz tienen botones y vistas generales que no se ven afectados por el *swipe* por diversas razones.

5.1. Descripción de la interfaz general

Para entender el funcionamiento de la interfaz general se muestra el siguiente diagrama:

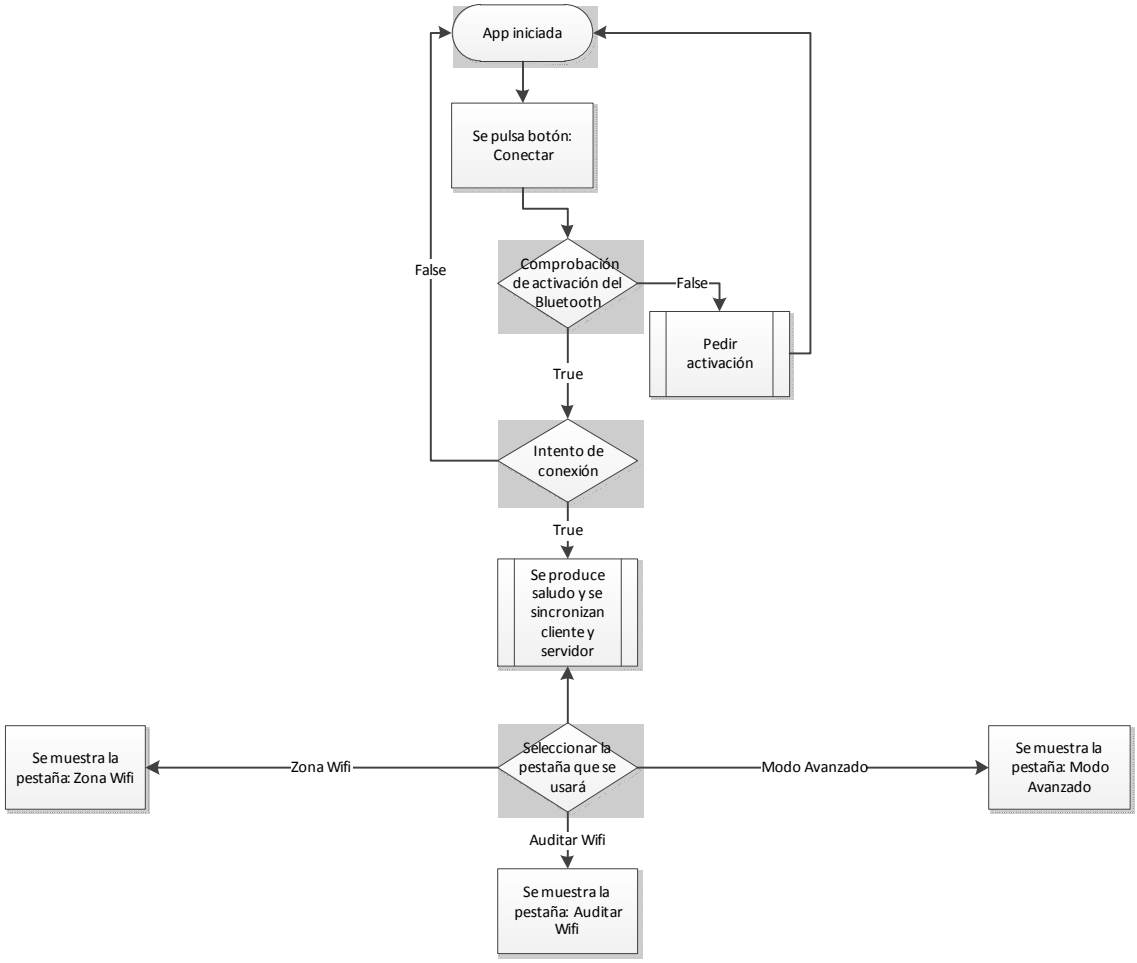

Ilustración 5: Diagrama de la interfaz general.

La interfaz general corresponde en la próxima imagen a los números del 1 al 4 y contiene lo siguiente:

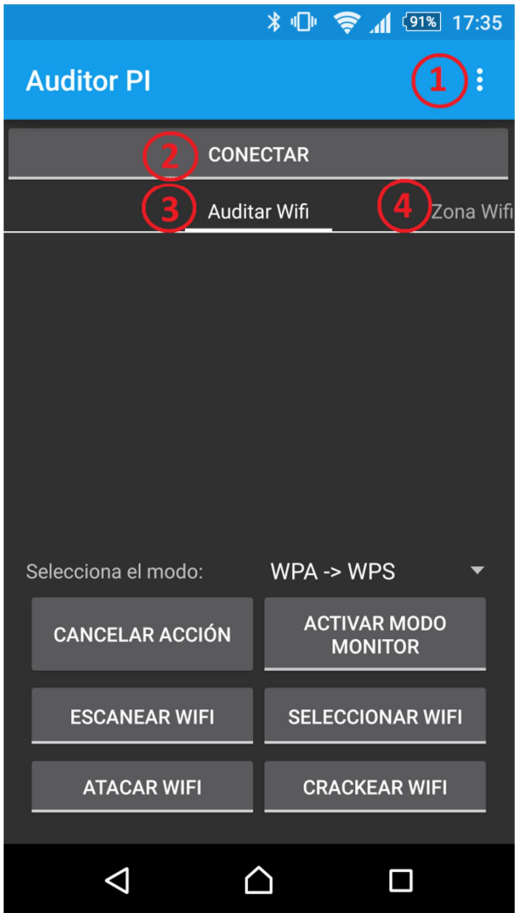

Ilustración 6: Descripción de la interfaz general.

1- Botón menú: Este botón es el predefinido por Android para mostrar el menú en Android. Este menú contiene las opciones siguientes:

Ilustración 7: Menú de la interfaz general.

1.1- Borrar terminal: Este botón limpia el terminal de salida, borrando así todo el texto que pudiera contener. El terminal de salida corresponde al número 5 en la imagen previa.

1.2- Reiniciar Raspberry Pi: Dicho botón reinicia la *Raspberry Pi*.

1.3- Apagar RaspBerry Pi: Esta opción apaga la la *Raspberry Pi*.

2- Botón *Conectar*: botón de tipo *ToggleButton* (Este es un tipo de botón de dos posiciones: *ON* y *OFF*). Al intentar presionar este botón, se comprobará si el Bluetooth del dispositivo Android está activo, y en caso de no estarlo se solicitará su activación. Dicho botón pasará a llamarse *Conectado* cuando esté activo. Este botón es general puesto que es imprescindible para todas las pantallas ya que se utiliza para conectar mediante Bluetooth con el servidor de la *Raspberry Pi*.

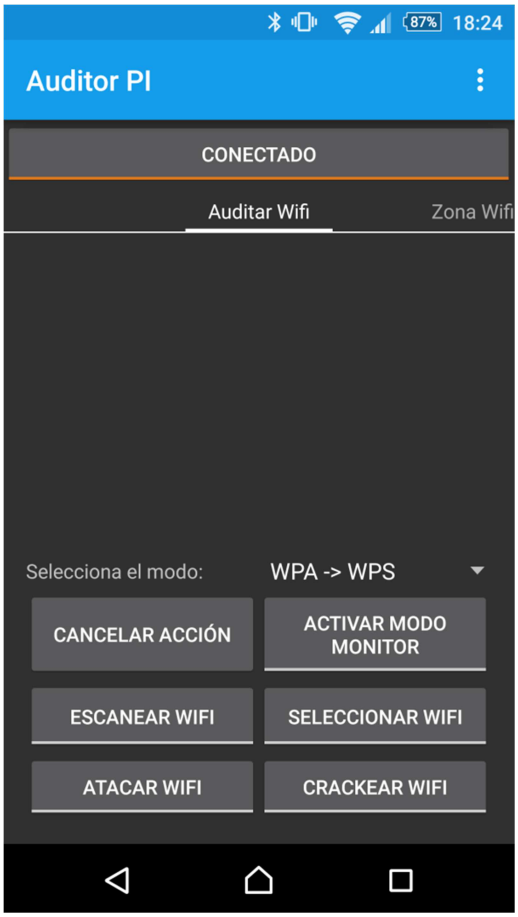

Ilustración 8: Descripción del botón conectar.

3- Auditar Wifi: Esto es la primera pestaña del *swipe*.

4- Zona Wifi: Esto es la segunda pestaña del *swipe*.

4.2-Modo Avanzado: Aunque en la imagen indicativa con los números no está visible. Este nombre coincide con la tercera pestaña del *swipe*.

Como se puede observar en la imagen de la interfaz con los números indicativos, en la pantalla inicial, en el *swipe* está como página o pestaña principal el modo *Auditar Wifi*.

5.2. Descripción de la interfaz Auditar Wifi

Para entender el funcionamiento de la interfaz Auditar Wifi se muestra el siguiente diagrama:

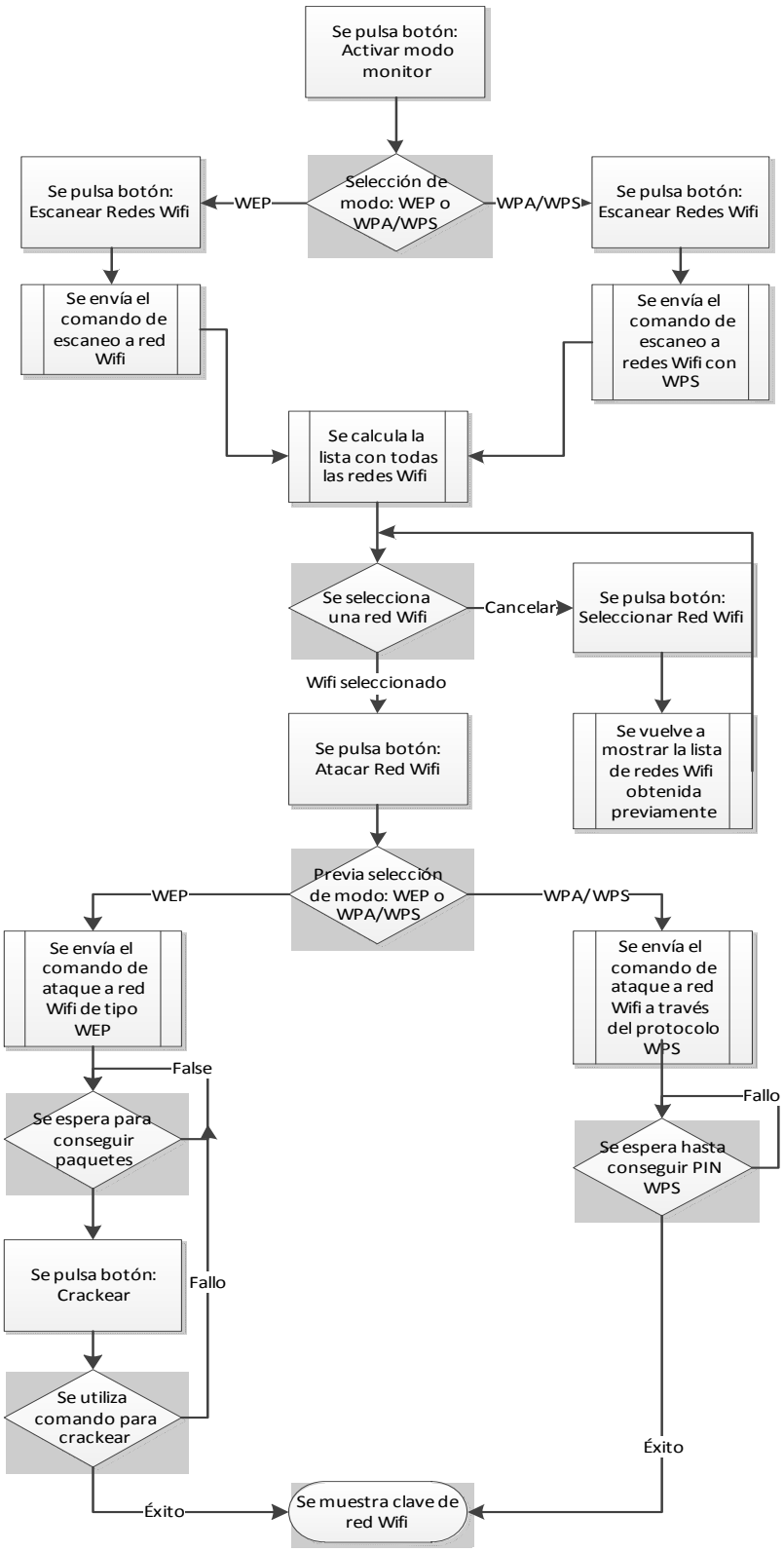

Ilustración 9: Diagrama de la interfaz Auditar Wifi.

En la pestaña *Auditar Wifi* hay lo siguiente:

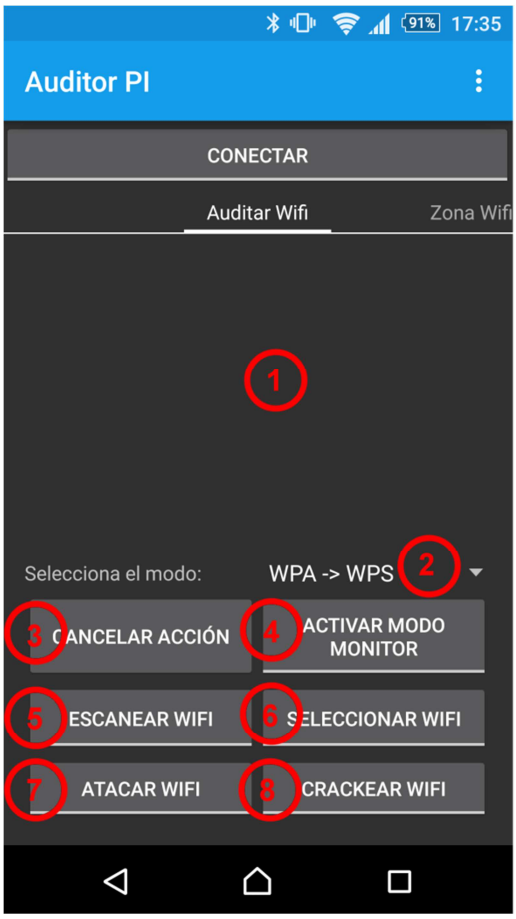

Ilustración 10: Descripción de la interfaz Auditar Wifi.

1- Terminal de salida: Implementado con un *TextView*. Esta vista ocupa la mitad del *fragment* (todo el contenido dentro de la pestaña *Auditar Wifi*), dejando la otra mitad al resto de botones y vistas. Esta vista muestra los resultados en bruto de los comandos bash que han sido enviados mediante Bluetooth. Esta vista utiliza un *slider* para subir y bajar para así no tener límite de texto. Aunque hay que destacar que dado que al tener una gran cantidad de texto puede dar problemas, se ha puesto un límite de líneas en 1000, de esta manera cuando se llegue a ese número de líneas la vista se limpia, evitando así sobrecargarla.

2- Selección de modo: Implementado mediante un *Spinner*. Se trata de un desplegable que muestra dos opciones:

2.1- WPA -> WPS: Esta es la opción por defecto. Si esta opción está activada predeterminará los comandos utilizados. Para el escaneo de redes Wifi se utilizará el *wash*. Dicho comando escanea únicamente las redes Wifi con cifrado WPA/WPA2 con el protocolo WPS. Para el ataque a una red Wifi utiliza el *reaver*. Este comando ataca a una red mediante el protocolo WPS, intentando obtener por fuerza bruta su PIN de 8 dígitos como ya se ha mencionado anteriormente. Y el botón *CRACKEAR WIFI* queda inhabilitado.

2.2-WEP: Como este cifrado a día de hoy apenas es usado, esta opción de ataque pasa a ser la secundaria. En este caso los comandos de escaneo y ataque de redes Wifi pasan a ser mediante *airodump-ng* y el crackeo utiliza la herramienta *aircrack-ng*.

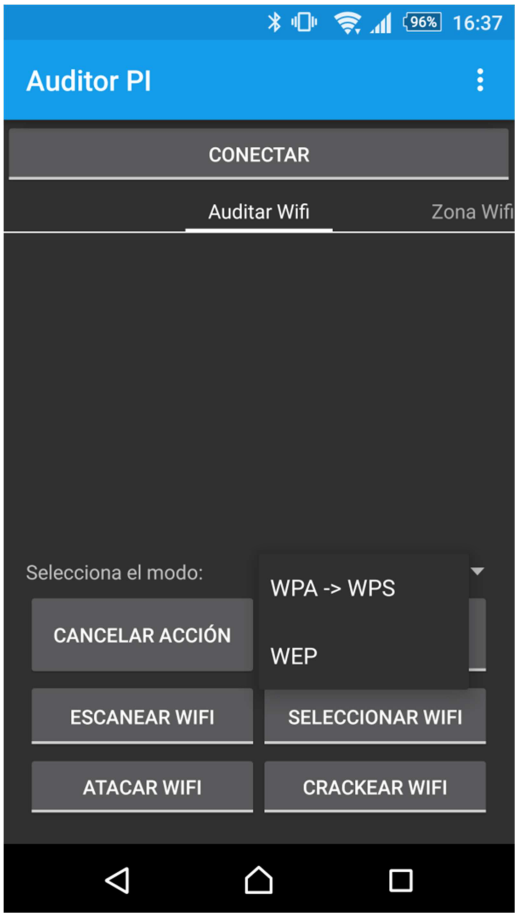

Ilustración 11: Spinner seleccionar modo en Auditar Wifi.

3- Cancelar acción: Botón implementado con un *Button*. Este botón envía un comando especial para cancelar el último comando ejecutado.

4- Activar modo monitor: Botón implementado mediante *ToggleButton*. Dicho botón, si está apagado, envía el comando *bash*: "airmon-ng start wlan0" y, si el *modo monitor* se inicia correctamente (comprobado mediante el análisis de la respuesta recibida), cambiará el texto a "Modo monitor activado". En caso de ya estar activo el botón (y por tanto el *modo monitor)*, se enviará el comando *bash*: "airmon-ng stop mon0", el cual desactiva el *modo monitor* y el botón vuelve a mostrar el texto "Activar modo monitor". Este botón sólo funcionará si se está conectado al servidor mediante Bluetooth; si no está conectado, saldrá un mensaje informando de que debe conectarse previamente.

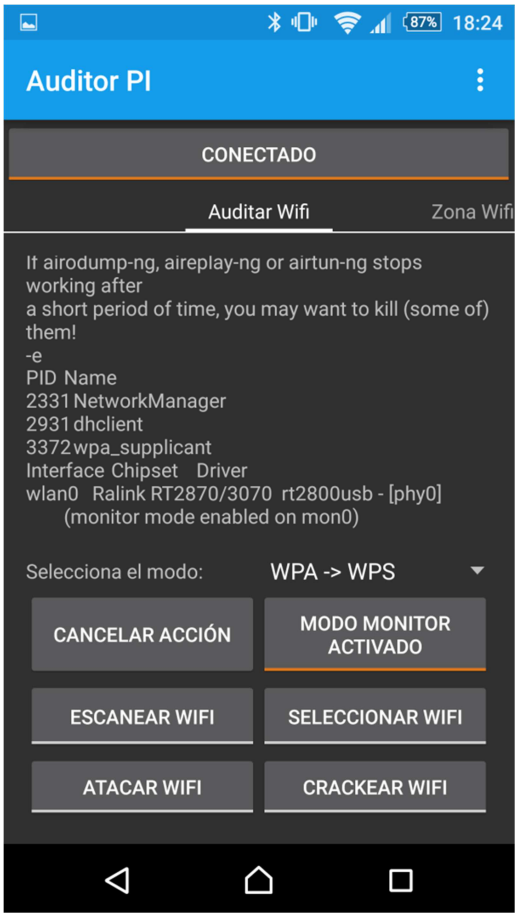

Ilustración 12: Modo monitor en Auditar Wifi.

5- Escanear Wifi: Botón implementado mediante *ToggleButton*. Cuando se utiliza dicho botón el texto cambiará a "Escaneando Wifi…", y volverá al texto "Escanear Wifi" cuando el escaneo termine. Además, este botón permite el *longClick*, es decir, este botón puede mantenerse presionado para acceder al menú de escaneo avanzado. Este botón solo funcionará en caso de estar conectado al servidor y tener el *modo monitor* activado, en caso contrario saldrá un mensaje indicando que el dispositivo no está conectado o que el *modo monitor* aún no está activo. Dependiendo del modo seleccionado (WPA -> WPS o WEP), los comandos a ejecutar serán los siguientes:

5.1-WPA -> WPS: Como ya se ha mencionado anteriormente se utilizará la herramienta *Wash* para realizar el escaneo de redes Wifi. Esta herramienta muestra en el escaneo únicamente las redes Wifi con el protocolo WPS activo, indicando asimismo el nombre de la red, MAC, canal, potencia con la que se recibe dicha red y si el protocolo WPS está bloqueado o no.

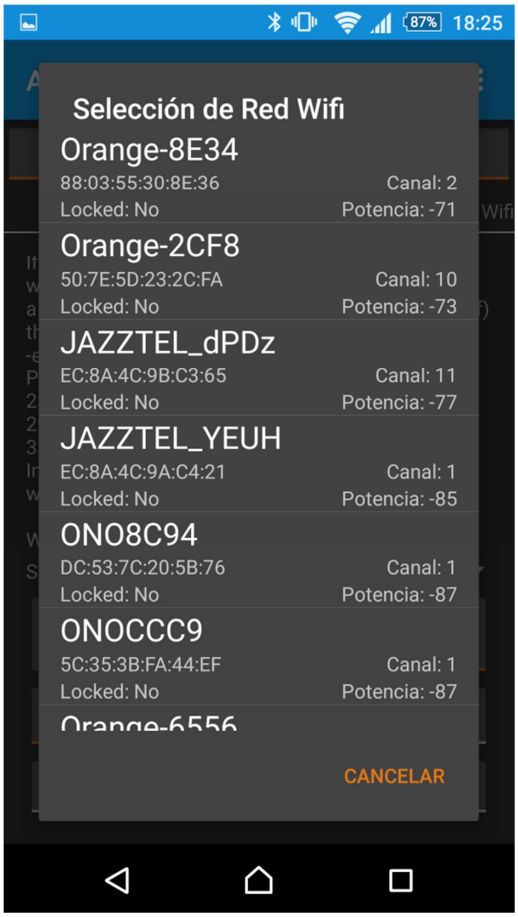

Ilustración 13: Escaneo WPS en Auditar Wifi.

5.1.1- El comando utilizado para el modo simple será: “wash -i mon0 -C” y se obtendrá una lista como la anterior, en la cual puedes deslizarte arriba y abajo mediante un *slider* para ver todas las redes Wifi escaneadas, que además están ordenadas según su potencia, estando más arriba cuanto mayor sea su potencia (medida en dBm). Al pinchar sobre una red de la lista, ésta se seleccionará activando el botón *Seleccionar Wifi*.

5.1.2- En caso de realizar un *longCLick* sobre el botón en lugar de ejecutar el comando previo, saldrá una ventana con la que cambiar los parámetros de la herramienta *Wash*. Esta ventana se implementa como un *Dialog*. En la parte de arriba se puede ver el comando por defecto, y mediante unos *RadioButton* se puede elegir si utilizar el modo *Survey* (por defecto) o el modo *Scan*. Además, se pueden introducir mediante un *EditText* parámetros personalizados como se ve en el ejemplo. Una vez confirmado el comando se enviará y se creará una lista con las redes Wifi como en el apartado anterior.

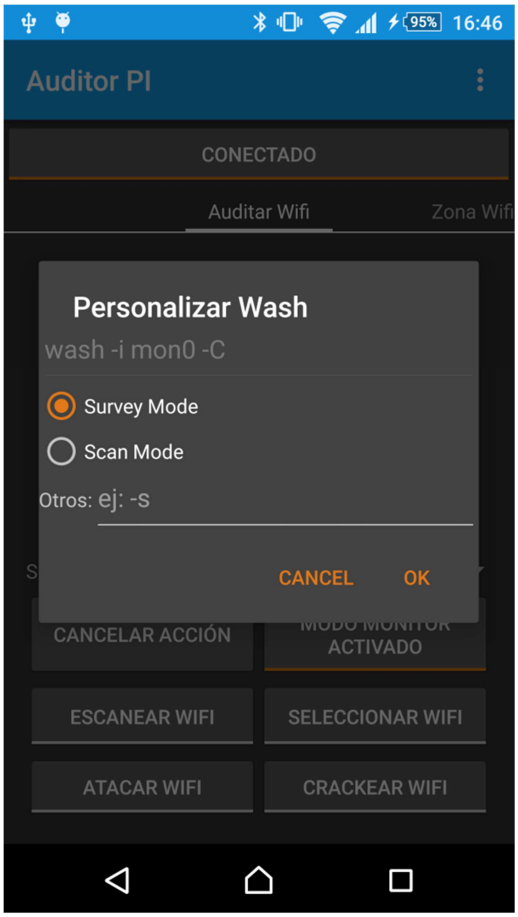

Ilustración 14: Personalización Wash en Auditar Wifi.

5.2-WEP: También se ha mencionado anteriormente que se utilizará la herramienta *airodump-ng* para realizar el escaneo de redes Wifi en el modo WEP. Esta herramienta muestra en el escaneo todas las redes Wifi, indicando asimismo el nombre de la red, MAC, canal, potencia con la que se recibe y el cifrado utilizado en dicha red.

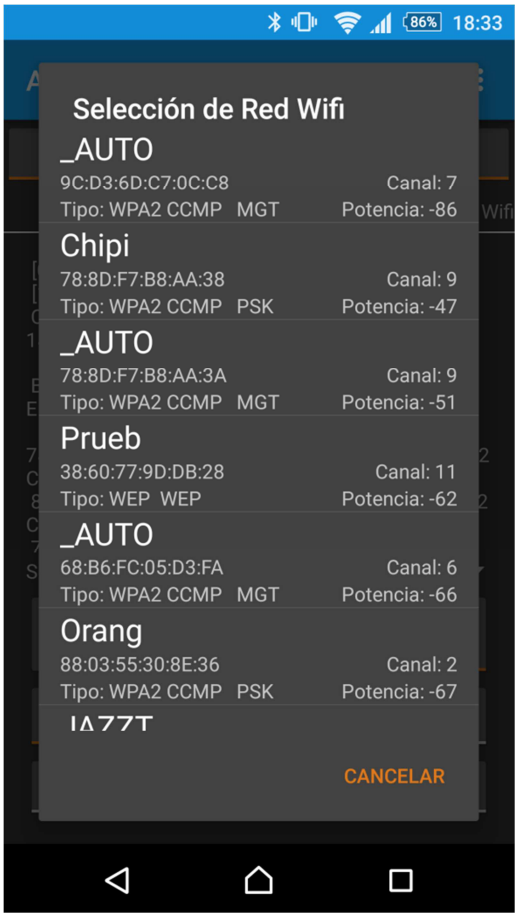

Ilustración 15: Escaneo WEP en Auditar Wifi.

5.2.1- El comando utilizado para el modo simple será: “airodump-ng mon0 -u 5” (se emplea una frecuencia de actualización de 5 segundos para no sobrecargar el sistema) y se obtendrá una lista como la anterior, en la cual puedes deslizarte arriba y abajo mediante un *slider* para ver todas las redes Wifi escaneadas, que además aparecen ordenadas según su potencia, estando más arriba cuanto mayor sea la potencia (medida en dBm). Al pinchar sobre una red de la lista, ésta se seleccionará activando el botón *Seleccionar Wifi*.

7.1.1- En caso de realizar un *longCLick* sobre el botón en lugar de ejecutar el comando previo, saldrá una ventana con la que cambiar los parámetros de la herramienta *airodump-ng*. Esta ventana se implementa como un *Dialog*. En la parte de arriba se puede ver el comando por defecto, y mediante *CheckBox* y *EditText* se pueden seleccionar y modificar los parámetros: *--update* (cambia la frecuencia de actualización de la lista, se mide en segundos), *-x* (establece el tiempo durante el cual se está escaneando, por defecto 0, que implica que la cancelación del escaneo será manual), *--channel* (Selecciona el canal o canales a escanear), *--encrypt* (indica el cifrado de las redes a escanear, el cual puede ser WEP, WPA y/o WPA2) y *-a* (que permite no mostrar redes que no tengan clientes activos). Además, se pueden introducir mediante un *EditText* parámetros personalizados como se ve en el ejemplo.

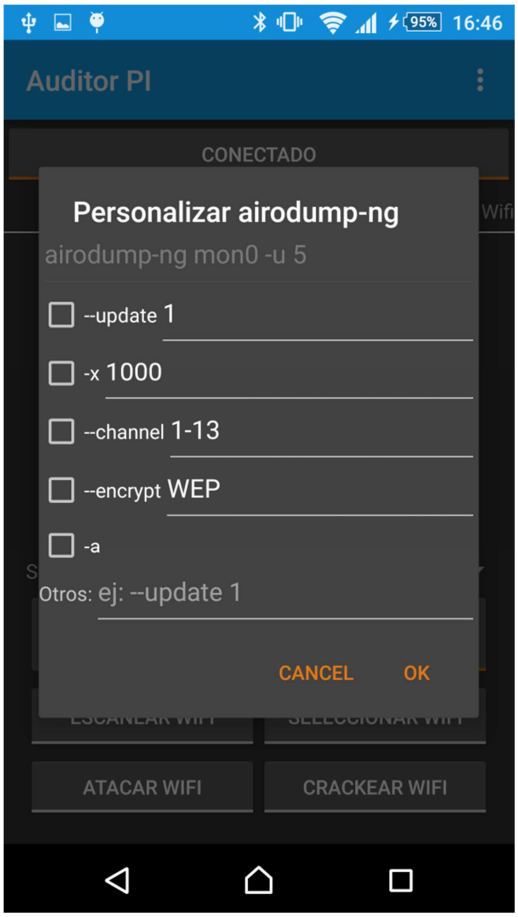

Ilustración 16: Personalización airodump-ng en Auditar Wifi.

6- Seleccionar Wifi: Botón implementado mediante *ToggleButton*. Este botón abrirá la lista generada en el botón *Escanear Wifi* para cambiar la red seleccionada de una manera mucho más rápida. Cuando una red Wifi sea seleccionada el texto de este botón cambiará a “Wifi seleccionado”, lo cual permitirá la realización de ataques y crackeos.

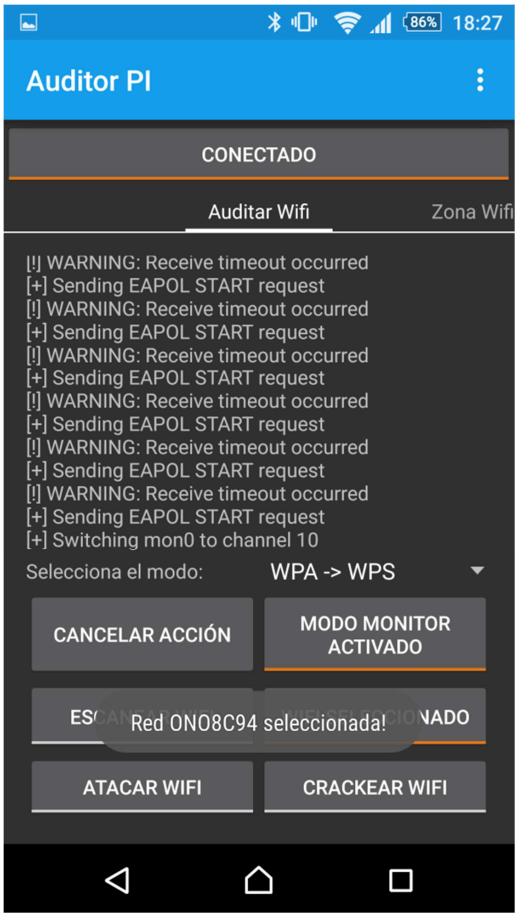

Ilustración 17: Selección Wifi en Auditar Wifi.

7- Atacar Wifi: Botón implementado mediante *ToggleButton*. Cuando se utiliza dicho botón el texto cambiará a "Atacando Wifi…", y volverá al texto "Atacar Wifi" cuando el ataque termine. Además este botón permite el *longClick* (aunque solo en el modo *WPA -> WPS*), es decir, este botón puede mantenerse presionado para acceder al menú de ataque avanzado. Este botón solo funcionará en caso de estar conectado al servidor, tener el *modo monitor* activado y haber seleccionado una red Wifi. En caso contrario saldrá un mensaje indicando si el dispositivo no está conectado, si el *modo monitor* aún no está activo o si falta por seleccionar la red Wifi. Dependiendo del modo seleccionado (WPA -> WPS o WEP), los comandos a ejecutar serán los siguientes:

7.1-WPA -> WPS: El ataque a redes Wifi con el protocolo WPS activo se realizarán mediante la herramienta *reaver* buscando así el PIN de 8 dígitos y probando en primer lugar una serie de números PIN por defecto. Asimismo cada vez que se prueba un PIN saldrá un mensaje en pantalla indicando el PIN actual.

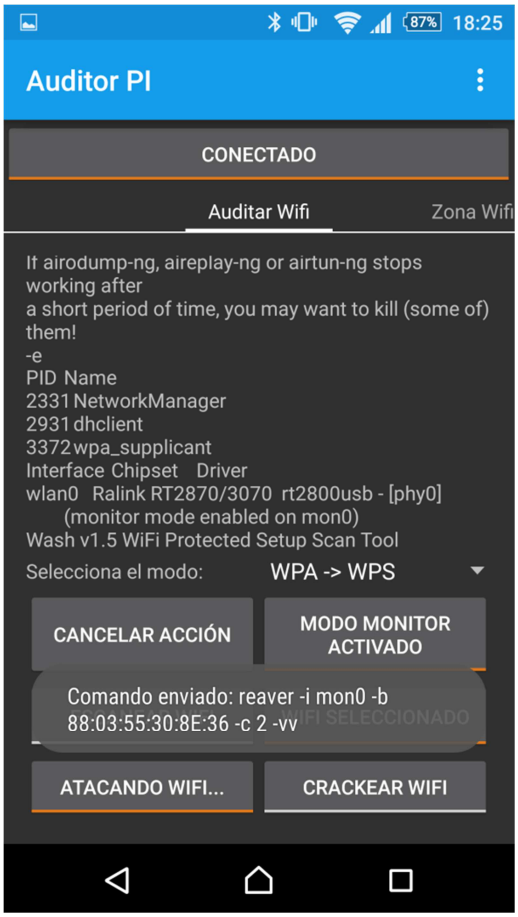

Ilustración 18: Ataque Wifi WPS en Auditar Wifi.

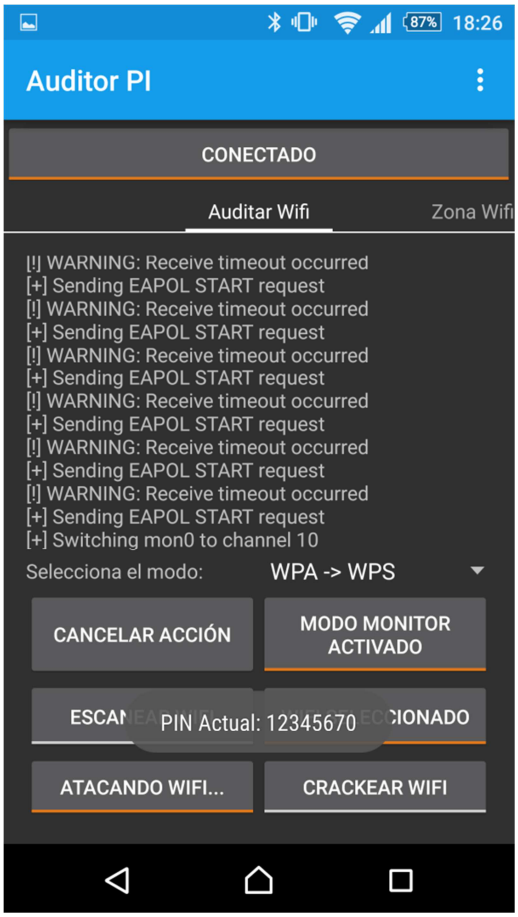

Ilustración 19: Probando PIN WPS en Auditar Wifi.

7.1.1- El comando utilizado para el modo simple será: “reaver -i mon0 -b MAC_RED_WIFI -c CANAL_RED_WIFI -vv”, donde la *MAC_RED_WIFI* y *CANAL_RED_WIFI* son las obtenidas previamente en la selección de la red Wifi de forma automática.

7.1.2- En caso de realizar un *longCLick* sobre el botón en lugar de ejecutar el comando previo, saldrá una ventana en la que es posible cambiar los parámetros de la herramienta *reaver*. Esta ventana se implementa como un *Dialog*. En la parte de arriba se puede ver el comando por defecto, y mediante *CheckBox* y *EditText* se pueden seleccionar y modificar los parámetros: *-d* (indica el tiempo entre el intento de dos números PIN), *--dh-small* (insta a utilizar intercambios de claves Diffie-Hellman más cortas pudiendo así acelerar el crackeo), *-T* (Establece el *timeout* de los intercambios M5 y M7 al número establecido, se mide en segundos), *-a* (auto-detección de las opciones avanzadas posiblemente más efectivas). Además, se pueden introducir mediante un *EditText* parámetros personalizados como se ve en el ejemplo.

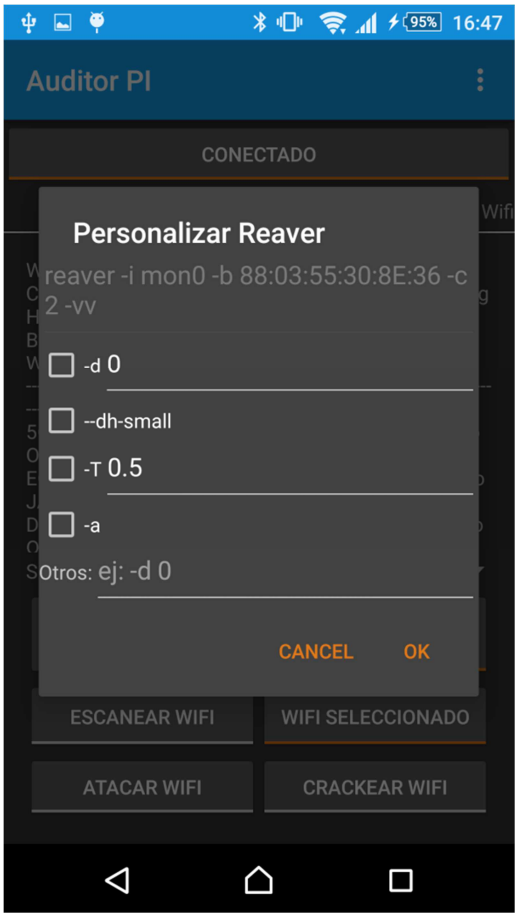

Ilustración 20: Personalizar Reaver en Auditar Wifi.

7.2-WEP: El ataque a redes Wifi con cifrado se realizarán mediante la herramienta *airodump-ng*.

7.2.1- El comando utilizado para el modo simple será: “airodump-ng mon0 -u 5 --bssid MAC_RED_WIFI -c CANAL_RED_WIFI -w captura”, donde la *MAC_RED_WIFI* y *CANAL_RED_WIFI* son las obtenidas previamente en la selección de la red Wifi de forma automática y “captura” indica el nombre del fichero **.cap* que se creará y almacenará los datos del ataque para el posterior crackeo.

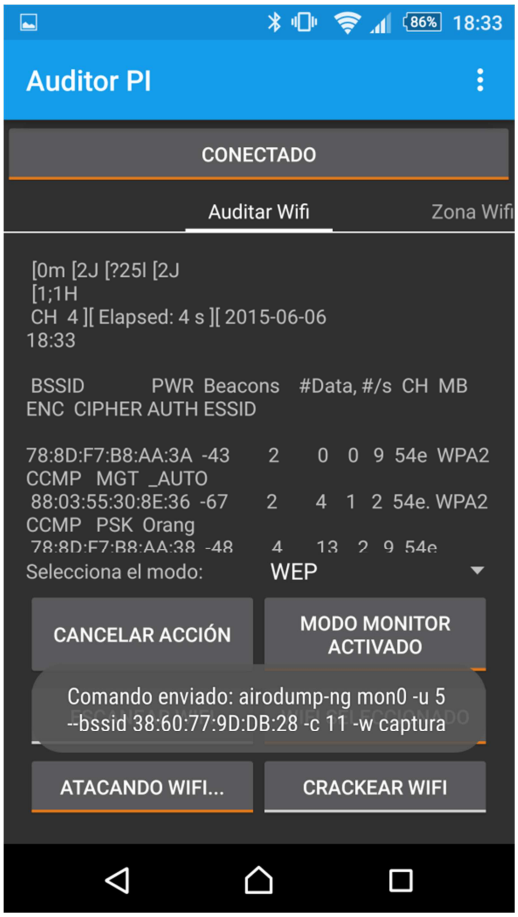

Ilustración 21: Ataque WEP en Auditar Wifi.

8- Crackear Wifi: Botón implementado mediante *ToggleButton*. Cuando se utiliza dicho botón el texto cambiará a "Crackeando Wifi…", y volverá al texto "Crackear Wifi" cuando el crackeo termine. Este botón solo funcionará en caso de estar conectado al servidor, haber seleccionado una red Wifi, y que esta tenga cifrado WEP. En caso contrario saldrá un mensaje indicando si el dispositivo no está conectado, si falta por seleccionar la red Wifi, o no tiene cifrado WEP.
El comando utilizado para el crackeo será: "aircrack-ng -a 1 -s captura*.cap -b *MAC_RED_WIFI*". Siendo la *MAC_RED_WIFI* son las obtenidas previamente en la selección de la red Wifi de forma automática, la *-a 1* indica que se trata de un cifrado WEP y *-s captura*.cap* es el fichero *.cap* en el cual se encuentra la información del ataque de la red Wifi y a partir de la cual se intentará crackear.

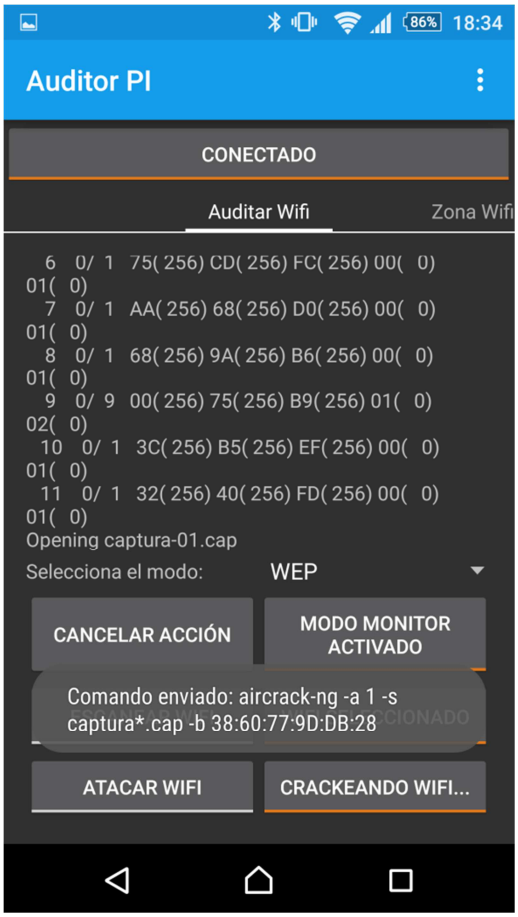

Ilustración 22: Crackeo WEP en Auditar Wifi.

5.3. Descripción de la interfaz Zona Wifi

Para entender el funcionamiento de la interfaz Zona Wifi se muestra el siguiente diagrama:

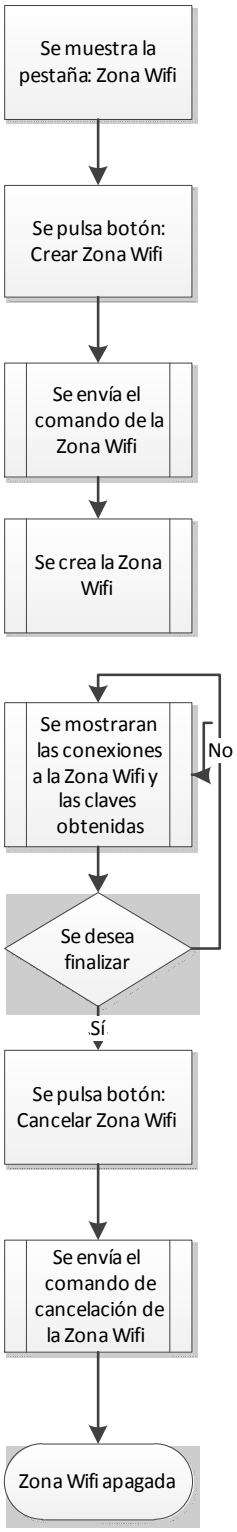

Ilustración 23: Diagrama de la interfaz Zona Wifi.

La pestaña *Zona Wifi* contiene lo siguiente:

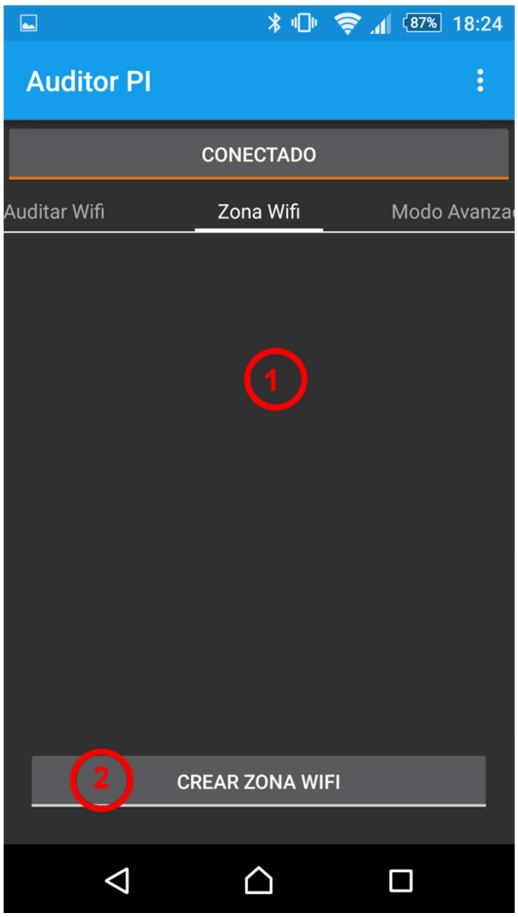

Ilustración 24: Descripción de la interfaz Zona Wifi.

1- Terminal de salida: Implementado con un *TextView*. Esta vista ocupa el 85% del *fragment* (todo el contenido dentro de la pestaña *Zona Wifi*), dejando el resto al botón. Esta vista muestra los resultados del script que se ha ejecutado. Estos resultados son, por una parte, la respuesta de la terminal según se vayan ejecutando las herramientas, y por otra parte, los clientes que se conecten a la red y las claves que se registren mediante el ataque *Man in the middle*. Esta vista utiliza un *slider* para subir y bajar para así no tener límite de texto. Hay que destacar que, dado que tener una gran cantidad de texto puede dar problemas, se ha definido un límite de líneas en 1000, ya que de esta manera cuando se llegue a ese número de líneas, la vista se limpia, evitando así sobrecargarla.

2- Crear Zona Wifi: Botón implementado mediante *ToggleButton*. Cuando se utiliza dicho botón el texto cambiará a "Zona Wifi creada", y volverá al texto "Crear Zona Wifi" cuando la Zona Wifi sea detenida en el momento en el que se vuelva a pulsar el botón. Este botón solo funcionará en caso de estar conectado al servidor; en caso de no estarlo, saldrá un mensaje indicando que el dispositivo no está conectado. Cuando se presione dicho botón se enviará el comando: "sh /root/Desktop/Fake_AP/kali-airssl-chipi.sh", el cual ejecutará el script *kali-airssl-chipi.sh*.

Este script utilizará las siguientes herramientas:

- airmon-ng: Para iniciar el modo monitor.
- airbase-ng: Para crear el AP falso.
- ifconfig, route e iptables: Para redirigir el tráfico de la red.
- dhcpd: Para utilizar DHCP en el AP falso.
- sslstrip: Para tratar de engañar al cliente con la finalidad de no usar SSL en la web.
- ettercap: Para realizar el *sniffing* o análisis paquetes.

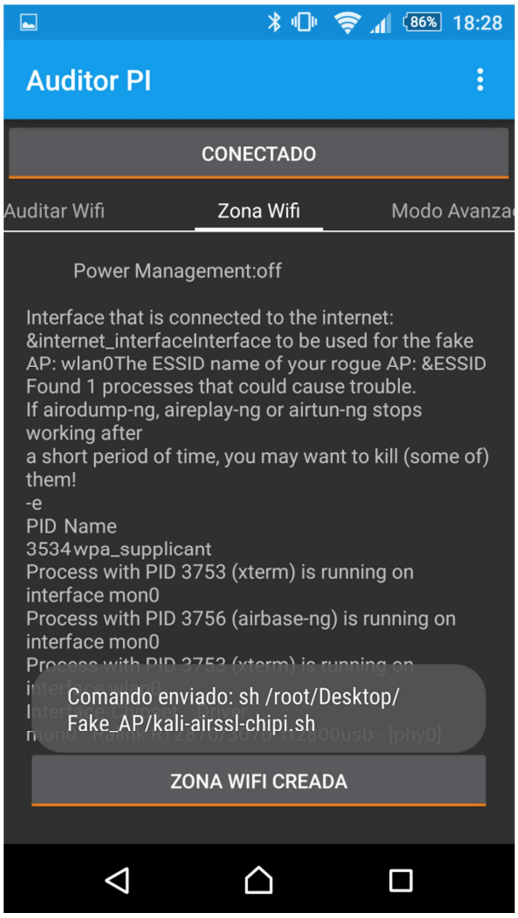

Ilustración 25: Creando una red en Zona Wifi.

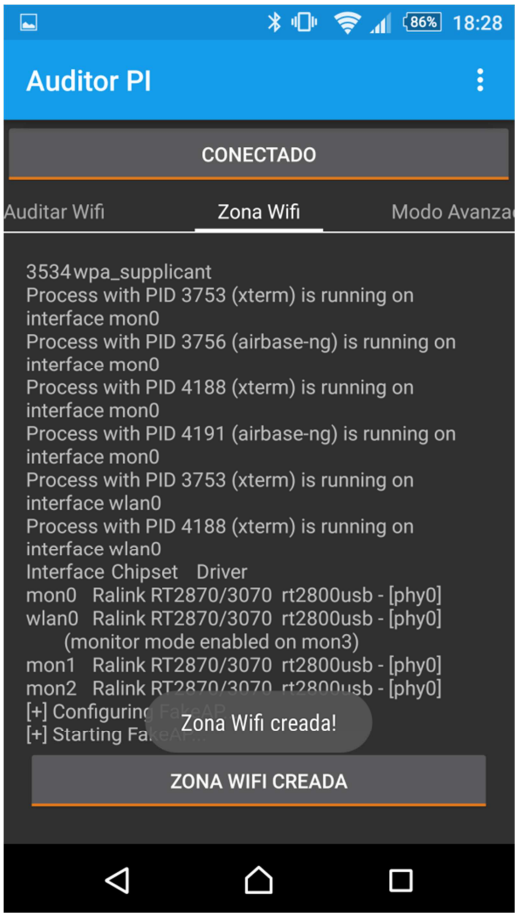

Ilustración 26: Red Wifi creada en Zona Wifi.

5.4. Descripción de la interfaz Modo Avanzado

Para entender el funcionamiento de la interfaz Auditar Wifi se muestra el siguiente diagrama:

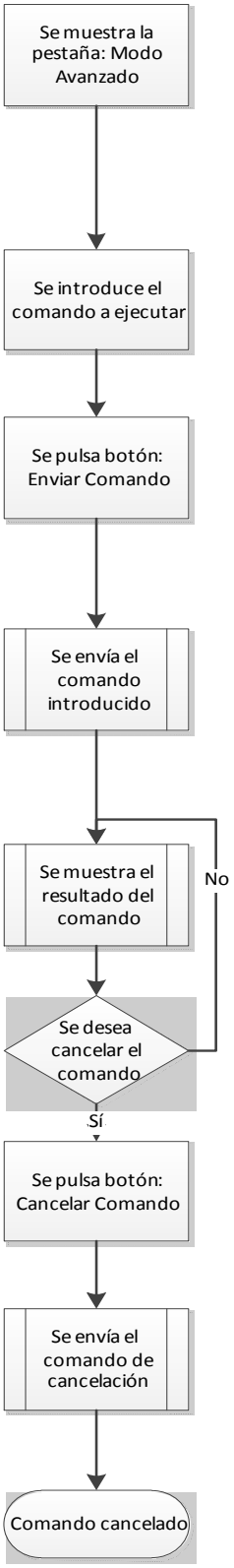

Ilustración 27: Diagrama de la interfaz Modo Avanzado.

La pestaña *Modo Avanzado* contiene lo siguiente:

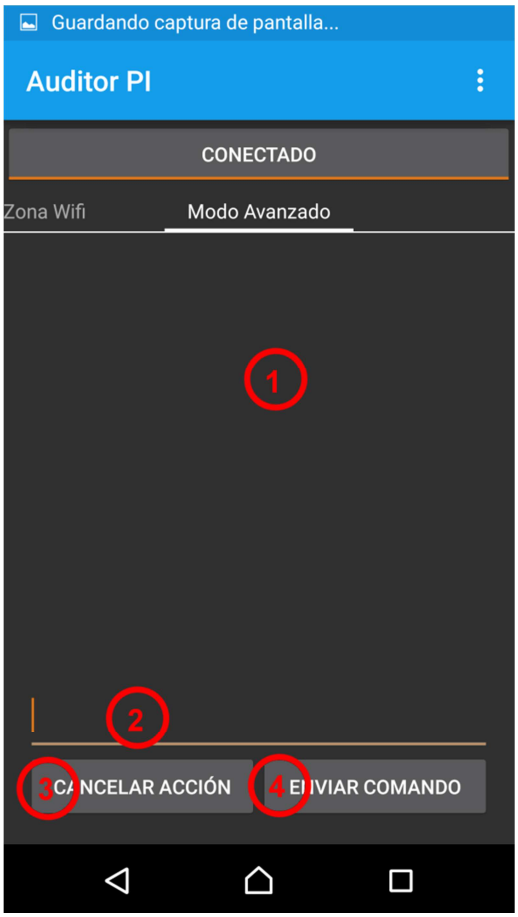

Ilustración 28: Descripción de la interfaz Modo Avanzado.

1- Terminal de salida: Implementado con un *TextView*. Esta vista ocupa el 70% del *fragment* (todo el contenido dentro de la pestaña *Modo Avanzado*). Esta vista utiliza un *slider* para subir y bajar para así no tener límite de texto. Igualmente hay que destacar que, dado que tener una gran cantidad de texto puede dar problemas, se ha definido un límite de 1000 líneas para que, cuando se llegue a ese número de líneas, la vista se limpie, evitando así sobrecargarla. Esta vista muestra los resultados en bruto de los comandos bash que han sido enviados mediante Bluetooth.

2- Entrada de terminal: Vista implementada mediante *EditText*. En este campo el usuario puede introducir los comandos *bash* que desee ejecutar en el servidor.

3- Cancelar Acción: Botón implementado mediante *Button*. Este botón solo funcionará en caso de estar conectado al servidor, y en caso de no estarlo saldrá un mensaje indicando que el dispositivo no está conectado. Su función es enviar un comando propio del programa para cancelar el último comando enviado al servidor.

4- Enviar Comando: Botón implementado mediante *Button*. Este botón solo funcionará en caso de estar conectado al servidor, y en caso de no estarlo saldrá un mensaje indicando que el dispositivo no está conectado. Cuando dicho botón sea presionado, si se ha escrito un comando en el campo *Entrada Terminal*, este será enviado al servidor y se

ejecutará como un comando *bash*, y su resultado se mostrará en la vista *Terminal de salida*.

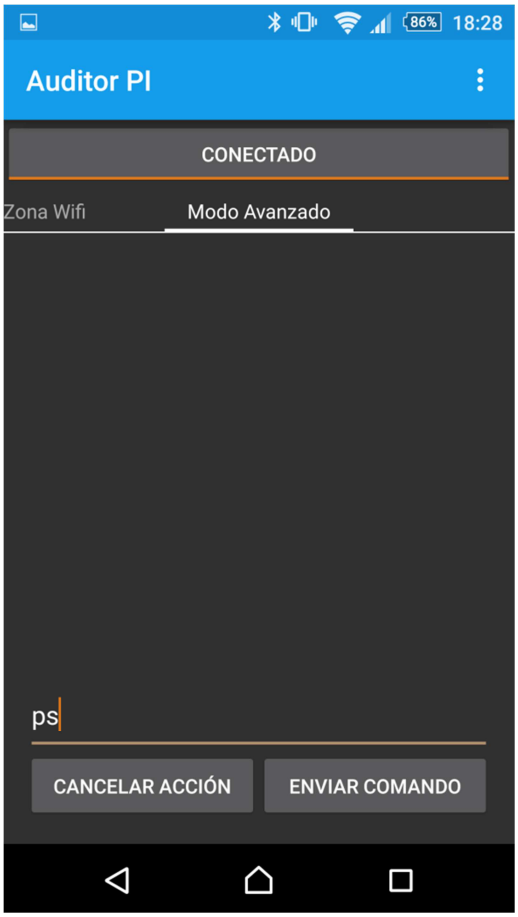

Ilustración 29: Introduciendo un comando en Modo Avanzado.

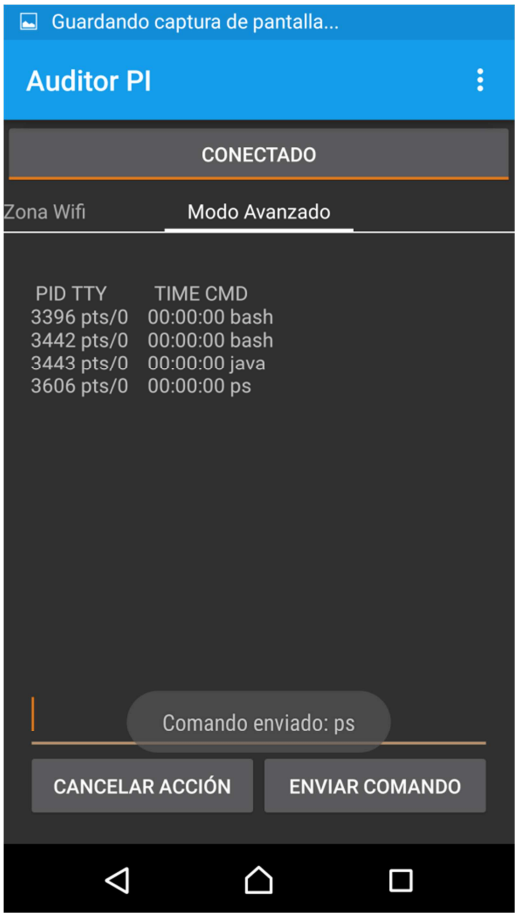

Ilustración 30: Comando enviado en Modo Avanzado.

6. Validaciones y pruebas

6.1. Ataques a redes Wifi

Se ha procedido a realizar ataques a redes Wifi de dos tipos: con cifrado WEP y con cifrado WPA/WPA2 mediante el protocolo WPS.

6.1.1- Ataque a red Wifi con cifrado WEP

Para esta prueba se ha utilizado un router *Cisco EPC3825*, tal como se ilustra en la figura de abajo.

Ilustración 31: Router Cisco EPC3825

Dicho router se ha configurado con un cifrado WEP y se ha establecido el nombre de la red como: “Pruebas”. Su clave ha sido generada automáticamente, y se corresponde con: “7FCF3147C3”.

Para tratar de obtener la clave se ha realizado el ataque estándar mediante el botón “Atacar Wifi”, el cual, utiliza el comando: “airodump-ng mon0 -u 5 --bssid MAC_RED_WIFI -c CANAL_RED_WIFI -w captura” (ver figura 32).

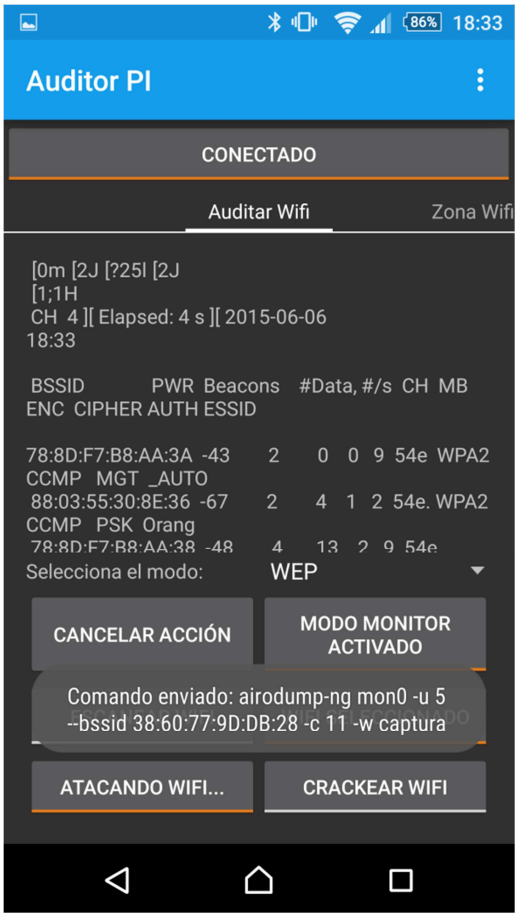

Ilustración 32: Atacando al AP "Pruebas".

Con este ataque se han ido obteniendo *IVs* (*Initialization Vector*) y, al llegar aproximadamente a los 25.000 *IVs*, se ha podido crackear la red Wifi con éxito utilizando el botón “Crackear Wifi”, el cual, utiliza el comando: “aircrack-ng -a 1 -s captura*.cap -b *MAC_RED_WIFI*” (ver figura 33).

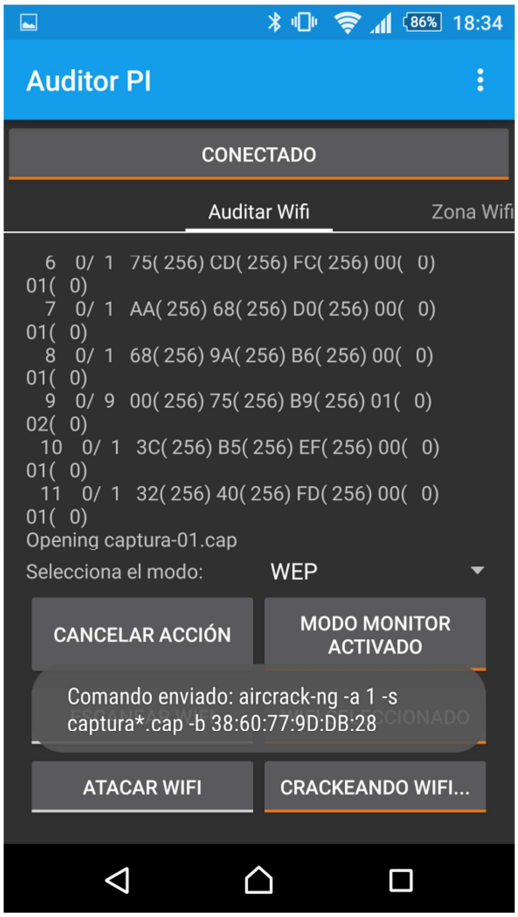

Ilustración 33: Crackeo de la red Wifi: "Pruebas".

Finalmente se ha obtenido la clave en aproximadamente 20 minutos, y ésta se muestra en la interfaz móvil (ver figura 34).

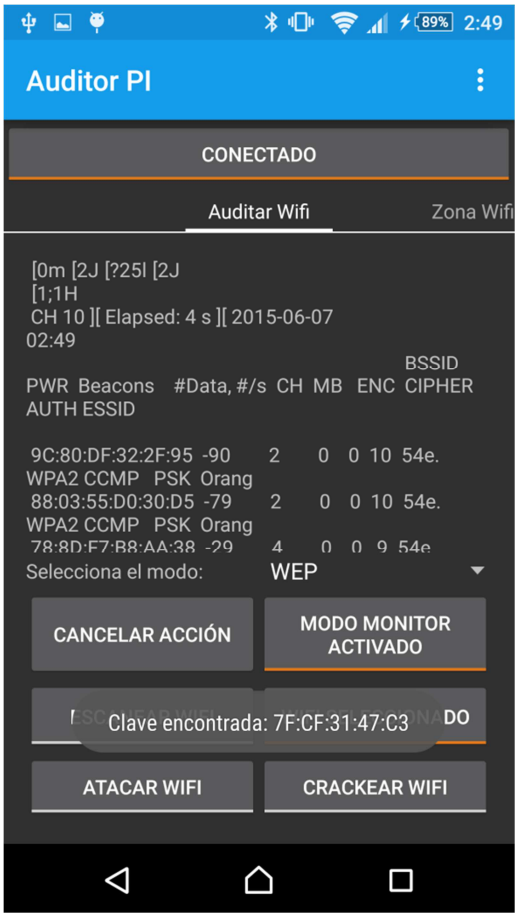

Ilustración 34: Obtención de clave WEP.

6.1.2- Ataque a red Wifi con cifrado WPA/WPA2 mediante protocolo WPS

Para este ataque, se ha utilizado el router *TP-LINK TL-MR3220*, el cual se ilustra en la figura de abajo.

Ilustración 35: Router TP-LINK TL-MR3220

Dicho router se ha configurado con WPA/WPA2 usando cifrado AES, y se ha establecido el nombre de la red como: “Chipi-Vodafone”. El PIN WPS se ha establecido aleatoriamente como “10527085”, y su clave ha sido establecida manualmente utilizando mayúsculas, minúsculas, números y otros caracteres, de forma que la contraseña sea compleja si se intenta obtener por diccionario. Dicha clave se corresponde con: “PrueBa123&3?+-”.

Para tratar de obtener la clave se ha realizado el ataque estándar mediante el botón “Atacar Wifi”, el cual, utiliza el comando: “reaver -i mon0 -b MAC_RED_WIFI -c CANAL_RED_WIFI -vv” (ver figura 36).

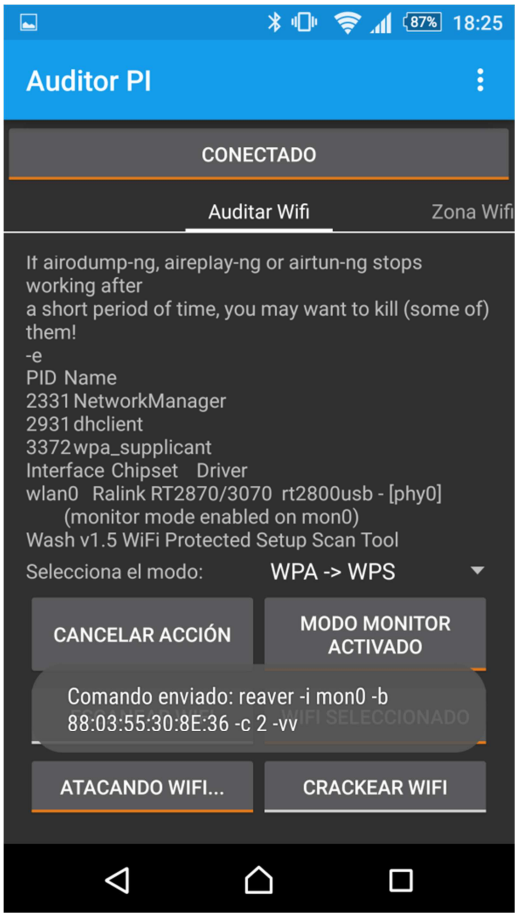

Ilustración 36: Ataque a red WPA/WPA2 mediante WPS.

En aproximadamente 5 horas se ha obtenido el PIN WPS, y con él la clave del AP, la cual se muestra en la aplicación móvil (ver figura 37).

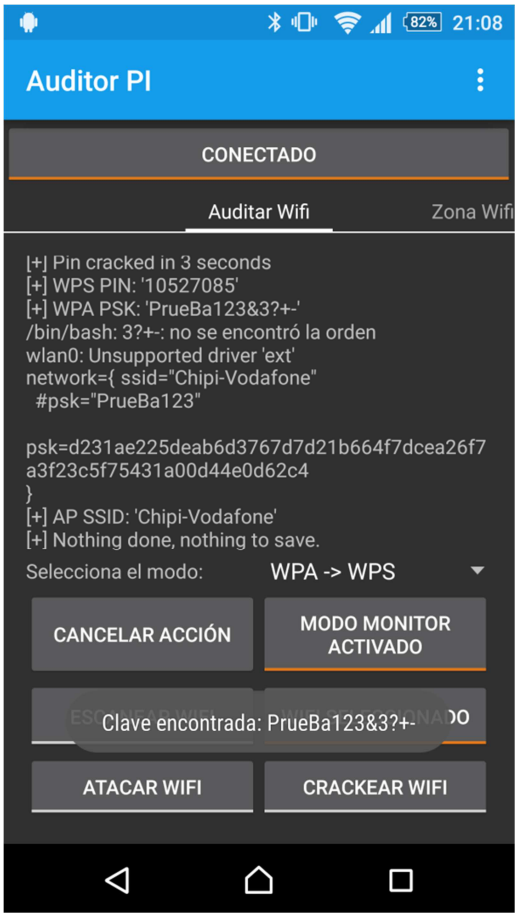

Ilustración 37: Clave WPA/WPA2 encontrada.

Además, se recibe la notificación *Push*, notificando asimismo la red Wifi que estaba siendo atacada, su PIN WPS y su clave (ver figura 38).

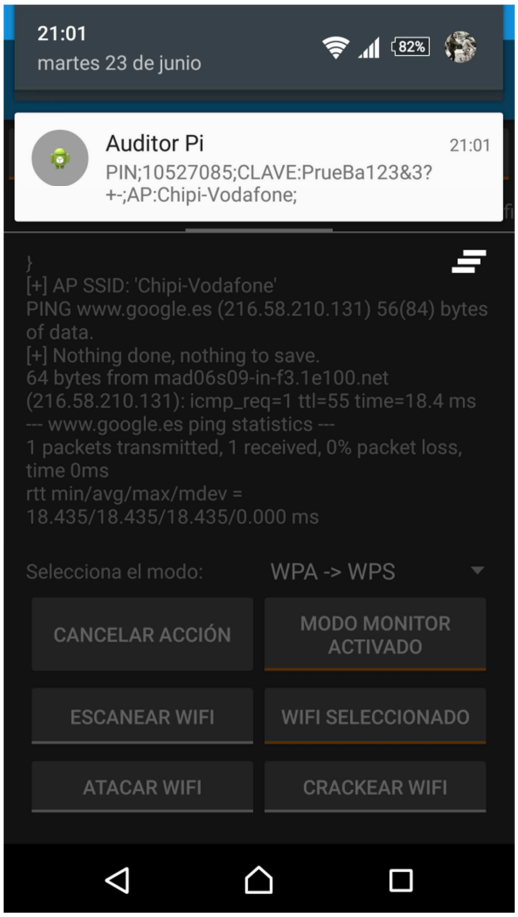

Ilustración 38: Notificación Push.

También se ha probado a estudiar el hecho de realizar ataques a redes Wifi mediante ataques de fuerza bruta.

Se ha utilizado la herramienta *pyrit*, concretamente se ha ejecutado la línea "pyrit benchmark". Esto hace un test de las claves por segundo que un computador es capaz de calcular para un ataque por diccionario, y además, marca la velocidad que proporciona cada núcleo del procesador.

En la *Raspberry Pi* se ha obtenido como resultado 80 claves por segundo, cuyo cálculo proviene de 20 claves por segundo por cada núcleo, teniendo éste 2 núcleos reales y 2 virtuales.

Un ordenador personal normal obtiene entre 2000 y 3000 claves por segundo, por lo que un ordenador personal puede calcular del orden de 30 veces más rápido. Además, si tenemos en cuenta que por ejemplo en un diccionario con palabras de 8 dígitos con caracteres numéricos obtenemos:

$10\ carácteres\ ^\wedge\ 8\ carácteres\ por\ palabra = 100.000.000\ claves\ posibles$

Para ordenador personal:

$$\frac{100.000.000\ claves\ posibles}{2.500\ claves\ por\ segundo} = 40.000\ segundos\ por\ diccionario$$

$$\frac{40.000\ segundos\ por\ diccionario}{3.600\ segundos\ en\ una\ hora} = 11\ horas\ por\ diccionario$$

Para *Raspberry Pi*:

$$\frac{100.000.000\ claves\ posibles}{80\ claves\ por\ segundo} = 1.250.000\ segundos\ por\ diccionario$$

$$\frac{1.250.000\ segundos\ por\ diccionario}{86.400\ segundos\ en\ una\ hora} = 14\ días\ por\ diccionario$$

Con estos cálculos se puede observar que el más sencillo de los diccionarios para WPA/WPA2 es únicamente factible mediante un ordenador personal, ya que con la *Raspberry Pi* el tiempo de cálculo comienza a ser bastante alto (dos semanas).

6.2. Consumo energético y duración de batería

Como se quería analizar el consumo para añadir la posibilidad de usar una batería, y así ver si sería posible realizar ataques con ésta haciendo de la *Raspberry Pi* un dispositivo móvil, se procedió a tomar datos de consumo.

Primeramente se probaron diferentes herramientas para analizar el consumo, como por ejemplo "PowerTop", e incluso las propias herramientas de Linux, pero resultó no ser posible puesto que, para analizar el consumo, se necesita una batería con sus drivers y era algo que se salía de lo pretendido, puesto que la idea sería poder usar incluso pilas para conseguir la autonomía del computador.

Finalmente se optó utilizar un medidor de consumo y se obtuvieron los siguientes datos:

MODO PASIVO

Raspberry Pi sin periféricos conectados: 1.3 Wh
Raspberry Pi con dispositivo Bluetooth sin utilizar: 1.3 Wh
Raspberry Pi con dispositivo Wifi sin utilizar: 2.0 Wh
Raspberry Pi con dispositivos Bluetooth y Wifi sin utilizar: 2.0 Wh

MODO ACTIVO

Raspberry Pi con dispositivo Bluetooth en uso: 1.4 Wh
Raspberry Pi con dispositivo Wifi en uso: 2.2 Wh
Raspberry Pi con dispositivos Bluetooth y Wifi en uso: 2.2 Wh
Raspberry Pi con dispositivos Bluetooth, Wifi en uso y crackeando (uso de CPU): 2.4 Wh

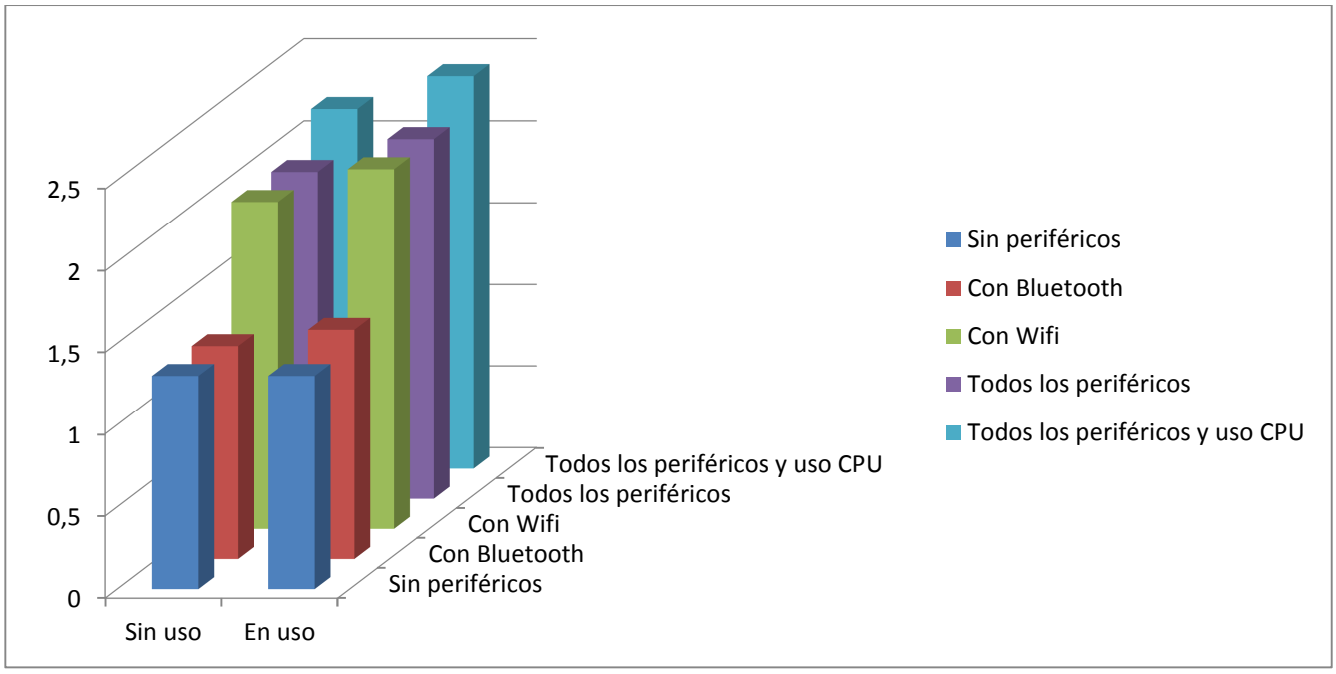

Ilustración 39: Gráfica consumo Raspberry Pi.

Ahora con una sencilla fórmula se pueden calcular los *mAh*:

$$Q(mAh) = E(Wh) \times 1000 / V(V)$$

Se va a suponer un consumo medio de 2.2 Wh debido a que al atacar una red Wifi, e independientemente de que estemos conectados mediante Bluetooth por el Android o no, el consumo será de 2.2 Wh. En el caso de que sea preciso crackear una red, se desprecia el consumo asociado, dado que el crackeo se produce en escasos segundos en comparación de los 3600 segundos a los que equivale una hora.

Entonces:

$$Q = 2.2Wh \times 1000/5V = 440mAh$$

Por tanto, se pueden plantear soluciones como son los *PowerBank*. Una solución recomendable por su relación coste-beneficio es el "Xiaomi PowerBank 10". Este banco de energía ofrece una capacidad de 10400 mAh por aproximadamente 15 €.

Ilustración 40: XiaoMi Power Bank

Por lo que:

$$\frac{10400mA}{440mAh} = 23.6\ horas$$

Así que es una muy buena opción ya que se obtiene una autonomía de aproximadamente 24 horas. Este tiempo es más que suficiente para obtener una clave ya sea por cifrado WEP o WPA/WPA2 si tiene el protocolo WPS activo. Cabe recordar que es factible romper el cifrado WEP en pocos minutos, mientras un ataque al protocolo WPS puede alargarse entre 6 y 10 horas. Esto significa que incluso se podrían hacer varios ataques sin recargar.

Por otra parte también es viable hacer una batería casera a base de pilas recargables AA o AAA, ajustándola a la capacidad deseada (Precio promedio de 1 € - 2 € los 1000 mAh).

7. Conclusiones

Se pretendía realizar una herramienta para analizar la seguridad y auditar redes Wifi de una manera sencilla utilizando para ello un sistema formado por un *Smartphone* o *Tablet* y un minicomputador como la *Raspberry Pi*, algo que hasta ahora no se encontraba desarrollado. Las únicas maneras de hacer esto eran mediante un *Smartphone* o *Tablet* con herramientas muy selectivas, difíciles de encontrar y poco eficaces, o mediante un ordenador personal con un sistema operativo enfocado a la Auditoría o seguridad Wifi. En este trabajo se ha conseguido realizar una fusión entre ambas posibilidades, otorgando así una mayor productividad, eficiencia y potencial.

Una vez concluido el desarrollo del proyecto, se tiene una interfaz de usuario atractiva, simple e intuitiva, la cual permite realizar ataques a redes Wifi de una manera simple, siempre y cuando se tengan unas nociones básicas de cómo funcionan las herramientas y los pasos básicos a seguir, como por ejemplo saber que es necesario activar el modo *monitor*. Para realizar ataques básicos a redes Wifi se usan los botones que implementan comandos predefinidos. Además su posible personalización la hacen sencilla de utilizar. Si atendemos también a la posibilidad de añadir una batería externa a la *Raspberry Pi*, logramos que este proyecto sea ligero y portátil.

Desde la perspectiva de un atacante, el principal inconveniente de este sistema es que no permite la realización de ataques de fuerza bruta dado a la reducida capacidad de cómputo de la *Raspberry Pi*. No obstante, destacar que en este proyecto se han conseguido alcanzar con éxito todos los objetivos definidos inicialmente.

Como posibles ampliaciones se podría aumentar el número de herramientas de acceso rápido, de forma a poder utilizarlas con un solo botón. También se podrían añadir nuevas funcionalidades a dicho proyecto, como por ejemplo permitir auditar los protocolos *Bluetooth* y *RFID*. Otra mejora podría ser crear un servidor web en el cual se muestren las estadísticas de la red Wifi mucho más detalladas y entendibles.

8. Bibliografía

[1] Instalación de Kali Linux en Raspberry Pi: http://docs.kali.org/kali-on-arm/install-kali-linux-arm-raspberry-pi

[2] Instalación sistemas operativos en Raspberry Pi: https://www.raspberrypi.org/documentation/installation/installing-images/windows.md

[3] Compilación de Bluecove para ARM: http://stackoverflow.com/questions/23142071/native-library-bluecove-arm-not-available

[4] APIs Android: https://developer.android.com/sdk/index.html

[5] Estilo visual de app Android: http://www.androidbegin.com/tutorial/android-viewpagertabstrip-fragments-tutorial/

[6] Desarrollo de la plataforma GCM: http://stackoverflow.com/questions/11294602/android-gcm-sender-id

[7] Desarrollo de la plataforma GCM: http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=push_android_gcloud

[8] Desarrollo de la plataforma GCM: http://hmkcode.com/android-google-cloud-messaging-tutorial/

[9] Conectarse a una red Wifi mediante terminal en Linux: http://www.blackmoreops.com/2014/09/18/connect-to-wifi-network-from-command-line-in-linux/

[10] Conectarse a una red Wifi mediante terminal en Linux: http://blog.desdelinux.net/como-conectarse-a-una-red-wifi-usando-el-terminal/

[11] Conectarse a una red Wifi mediante terminal en Linux: https://mislinuxapps.wordpress.com/2010/03/10/como-conectarse-a-una-red-wifi-mediante-terminal/

[12] Conectarse a una red Wifi mediante terminal en Linux: http://maztories.blogspot.com.es/2013/05/como-conectar-una-red-wifi-usando-la.html

[13] Instalación de PowerTop: https://01.org/powertop/

[14] Instalación de ncurses para PowerTop: http://geeksww.com/tutorials/operating_systems/linux/tools/how_to_download_compile_and_install_gnu_ncurses_on_debianubuntu_linux.php

[15] Instalación de la nueva versión de la herramienta *reaver*: https://code.google.com/p/reaver-wps-fork/

[16] Ataque a protocolo WPS offline: https://www.wifi-libre.com/topic-87-pixiewps-de-wiire-la-herramienta-para-el-novedoso-ataque-pixie-dust.html

[17] Manual para la herramienta *sslstrip*: http://www.redeszone.net/seguridad-informatica/sslstrip/

[18] Descarga y manuales de la herramienta *intercepter-ng*: http://sniff.su/

[19] Crackear redes WEP – Pedro Escribano: http://www.pedroescribano.com/docs/Crackear_redes_WEP.pdf

[20]James F. Kurose & Keith W. Ross (2011), “Redes de Computadoras: un enfoque descendente”, 5ta Edición de Prentice Hall
[21]Pablo González Pérez (2013), “Pentensting con Kali” de 0xWord
[22]Wei Meng Lee (2012), “Android 4 : desarrollo de aplicaciones” de Anaya