WiFi-Pocket.

Herramienta hardware para RedTeam y BlueTeam en redes WiFi.

Hardware and software tool for Red Team and Blue Team

Trabajo de Fin de Grado
Curso 2019–2020

Autor
Daniel García Baameiro

Directores
Inmaculada Pardines Lence
Marcos Sánchez-Élez Martín

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

WiFi-Pocket.

Herramienta hardware para RedTeam y BlueTeam en redes WiFi.
Hardware and software tool for Red Team and Blue Team

Trabajo de Fin de Grado en Ingeniería Informática

Departamento de Arquitectura de Computadores y Automática

Autor
Daniel García Baameiro

Directores
Inmaculada Pardines Lence
Marcos Sánchez-Élez Martín

Convocatoria: *Enero 2020*

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

23 de enero de 2020

Autorización de difusión

El abajo firmante, matriculado en el Grado de Ingeniería en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado: "WiFi-Pocket. Herramienta hardware para RedTeam y BlueTeam en redes WiFi", realizado durante el curso académico 2019/2020 bajo la dirección de Inmaculada Pardines Lence y Marcos Sánchez-Élez Martín en el Departamento de Arquitectura de Computadores y Automática, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Daniel García Baameiro

23 de enero de 2020

Dedicatoria

A mi abuelo, por haber sido una de las personas más importantes y que más me han marcado en la vida. Por haberme enseñado a luchar por mis metas y a perseguir mis objetivos, sin rendirme. Por haberme preparado, como bien tú mismo decías , para "darle caña" a la vida.

Gracias por todo lo que me has dado, por esas partidas a las damas donde empecé siendo un mero aprendiz y acabé ganando al maestro, por haber sido quien comenzó a saciar mi curiosidad con conceptos tan simples como la escala musical, por ser quien me enseñó a subirme a una bici sin miedo a las caídas, sin miedo a lo que pudiera pasar, superando el dolor, siguiendo hacia adelante sin mirar atrás.

Gracias por confiar en mí, tu nieto. Creo que nunca podré agradecerte lo suficiente todo lo que me has dado.

A mis padres, por haber forjado en mí la persona que soy ahora y por todo el apoyo que me han dado desde el comienzo de mi camino.

A mi hermano, porque pese a nuestras peleas ha estado todo el tiempo ahí, para lo que he necesitado, cuidándome incluso desde la distancia.

A mis primas, quienes han construido en mi memoria los mejores recuerdos de mi vida, pues he podido llegar incluso a vivir aventuras en barcos imaginarios construidos con palés de madera.

A mi mejor amigo, la persona que me ha acompañado desde que era un simple crío y que sigue estando ahí.

A mi familia, por haber sido uno de los pilares que conforma mi persona.

Agradecimientos

Principalmente, me gustaría dar las gracias a mis directores del TFG por el tiempo y la ilusión que me han dedicado y que han empleado en el presente proyecto.

Gracias a todos los profesores que confiaron en mí y que me han permitido llegar hasta aquí. Gracias a esos profesores que han marcado mi enseñanza, desde primaria con Aida, en la ESO con Hermila y con Juan Luis, en bachiller con Cesar y en la universidad con Raquel. Sin duda sois los sabios que han aportado su granito de arena para que pudiera aprender más y crecer como persona.

Gracias a los dos motores de motivación que han marcado mi etapa universitaria.

Por un lado, a Pablo Moreno Ger, que depositó en mí su confianza y vió en mí la capacidad para dar ponencias en la universidad cuando él era Vicedecano de Innovación. Pese a ser yo un alumno, me recibía en su despacho, nos sentábamos en la mesa, lanzábamos lluvia de ideas y pensábamos la mejor forma para planificar los proyectos que se tenían desde LibreLabUCM y que queríamos realizar en la Facultad. Dice mucho de una persona que te trata como a un igual, y no como a un simple alumno.

Por otro lado, a Raquel Hervás Ballesteros, quien ha sido una de las profesoras con las que he obtenido mis mejores calificaciones en el Grado de Ingeniería Informática. Siempre quedarán esas ganas de matrícula de honor. Cuando una profesora lucha por un alumno, se preocupa por que aprenda, y consigue enseñar más conocimiento aparte de los impartidos en las clases solo por la curiosidad del alumno, es cuando estamos ante una gran profesora.

No quiero dejarme a todas aquellas personas que han ayudado a que todos los proyectos en los que me he visto involucrado y que han aportado parte de lo que soy como persona.

Gracias a Carol y Juan Luis, de La Casa, por haberme dado soporte y ayuda en todos los problemas que han acontecido.

Gracias a los compañeros que conocí a principio de curso, Marta, Guillermo y Carlos, por darle otra visión a la carrera.

Gracias a Keto, por toda la confianza que ha depositado en mí.

Gracias a Álvaro, por su "moral support" y por ver en mí la capacidad para ser mejor.

Gracias a mis compañeros de LibreLabUCM, tanto de generaciones pasadas constituidas por Antonio, Samer, Abel, Jesús y Sem, como de las nuevas generaciones. Gracias también a mis compañeros de otras asociaciones. Al final, las generaciones primeras que vimos surgir LibreLabUCM contemplamos orgullosas a las nuevas conformadas por Rafa, Cristóbal, David, Pedro, Luis, ... Siempre nos quedarán todas las anécdotas o los momentos de cacharreo con Pablo y Miguel Ángel. Gracias a todos los socios que conforman LibreLabUCM, asociación forjada con mucha ilusión para los presentes y futuros alumnos de la Universidad Complutense de Madrid.

Gracias a mi compañera Míriam, con quien he podido dedicar parte de mi tiempo en la universidad y que me ha acompañado por caminos duros y difíciles que se han conseguido sobrepasar.

Gracias a Adriano, quien me ha donado parte de las sondas que se han utilizado para implementar y probar el presente Trabajo de Fin de Grado.

Finalmente, dar las gracias a todos aquellos hackers que comparten la cultura libre y que publican sus conocimientos de forma totalmente gratuita a través de la red.

Vivimos en un mundo conectado, un mundo conferido a lo virtual, sin barreras, sin fronteras. Un mundo que nos permite intercambiar información con distintas personas a lo largo del planeta.

Vivimos en un mundo donde la tecnología crece a un nivel exponencial, donde las nuevas generaciones han tomado el relevo de las antiguas. Un mundo que se avecina nuevo, un mundo donde cada gota de tiempo llamada segundo es un nuevo descubrimiento.

Vivimos en un mundo... ¿Qué se avecina seguro?

Daniel García Baameiro

Resumen

Todos los hogares y la mayoría de empresas cuentan con conexiones a internet vía WiFi. A través de estas comunicaciones se envían datos privados de alto valor. La seguridad de estas redes es frágil y, en la actualidad, suelen ser uno de los principales vectores de ataque de los ciberdelincuentes.

El objetivo de WIFi-Pocket es conseguir que los usuarios puedan construirse su propio set multiusos de pentesting a través de herramientas hardware y software.

Por un lado, en la parte defensiva, se busca poder identificar en tiempo real ataques a nuestra infraestructura de red inalámbrica WiFi. Por otro lado, en la parte ofensiva, podremos identificar mediante pruebas de intrusión las vulnerabilidades de nuestra infraestructura. Todo ello a través de una interfaz web responsive amigable con el usuario y accesible a través de cualquier dispositivo que cuente con un navegador web.

Palabras clave

Ciberseguridad, Hacking, Blue Team, Red Team, WiFi, Auditoría, IDS, WIDS, Pentesting, Hardware

Abstract

Many homes and companies have internet connections via WiFi. Through these communications, high-value private data is sent. Nowadays the security of these networks is so fragile and they are usually one of the main attack vectors of cybercriminals.

The objective of WIFi-Pocket is to ensure that users can build their own multipurpose set of pentesting through hardware and software tools.

On the one hand, on the defensive side, we seek to identify real-time attacks on our WiFi wireless network infrastructure. On the other hand, in the offensive part, we will be able to identify through vulnerabilities tests the vulnerabilities of our infrastructure. All this through a responsive web interface user friendly and accessible through any device that has a web browser.

Keywords

Ciberseguridad, Hacking, Blue Team, Red Team, BlueTeam, RedTeam, WiFi, Auditoría, IDS, Pentesting, Hardware

Índice

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

"Internet es la forma de expresión más emocionante, moderna y desarrollada que existe. La expresión es un derecho humano. Claro que internet también tendría que serlo."
— Aaron Swartz

1.1. Objetivos

Este proyecto tiene como objetivo principal poder dotar a los equipos tanto de Blue Team como de Red Team de las herramientas necesarias para facilitar su labor.

Como objetivo secundario, pretende que cualquier usuario pueda hacer uso del proyecto independientemente de sus conocimientos para poder proteger su infraestructura de comunicaciones.

Otro de los objetivos que se cumpliría de forma paralela a los mencionados anteriormente es la labor de mantener a la población segura de ataques externos, que puedan suponer un perjuicio para la sociedad.

WiFi-Pocket consiste en una suite de herramientas orientadas a tecnología WiFi haciendo uso o no de sondas y de un panel de control. Todo ello de una forma fácil y simplificada, orientada al uso de cualquier usuario.

1.2. Motivación

Actualmente, y cada vez más en aumento, la sociedad tiende hacia el uso de redes inalámbricas de tipo WiFi. Estas redes permiten al usuario conectarse a Internet sin la necesidad de realizar una conexión física, por cables.

El acceso a estas redes se hace a través de un punto de acceso. Este punto de acceso si bien puede requerir de una contraseña para navegar a través de él, también puede estar abierto. Estos puntos de acceso sin contraseña comunmente pertenecen a comercios o locales que requieren de datos privados del usuario si este quiere conectarse a ellos.

El intercambio de información entre un usuario y un punto de acceso se hace

de forma pública, es decir, la información es enviada en forma de paquetes a través de ondas donde cualquier persona puede capturarlas. Esta captura no implica que el usuario final se quede sin el paquete, simplemente se produce una lectura del mismo por parte de un usuario externo. El mero hecho de interceptar paquetes no requiere de ninguna conexión con el punto de acceso, y no dispone de registro que indique que se está realizando tal actividad. Se trata de un acto sin repercusión que no puede ser detectado.

Las redes protegidas añaden una capa de cifrado a los paquetes impidiendo que se pueda saber el contenido del paquete o que información está transportando. Si una persona quisiera extraer dichos datos, necesitaría tener acceso a las medidas de seguridad implementadas por el punto de acceso.

Un atacante podría explotar vulnerabilidades presentes en las capas de cifrado aplicadas sobre los paquetes enviados entre los dispositivos conectados a un router o hacer uso de trucos de ingeniería social con el fin de atacar a los usuarios de una red. Estos usuarios podrían no ser conscientes de que les están atacando o bien podrían haber sido ya atacados y no haberse dado cuenta. El objetivo de estos ataques tiende a ser robar información personal o obtener total acceso sobre los dispositivos de los usuarios.

Estos ataques pueden aplicarse no solo a usuarios individuales sino también a grandes infraestructuras. Esto es debido a que, en la actualidad, no hay infraestructura que no dependa de la implementación de un dispositivo vía WiFi para sus usuarios y/o empleados.

Se propone en este Trabajo de fin de grado la creación de una herramienta que permita emular ser un atacante a la par que permita detectar en tiempo real ataques que se realicen sobre una infraestructura de comunicaciones inalámbrica vía WiFi. Para la creación de esta herramienta, se necesitará volcar el software sobre varios dispositivos hardware que se denominarán sondas conectadas a un panel web central. La información que intercambian se realizará sobre su propia red virtual, evitando que cualquier persona pueda acceder a la información.

1.3. Motivación personal

La curiosidad por la ciberseguridad comenzó en mi vida el día que llamó mi atención a los 12 años, estando yo en 1º de la ESO. Empecé a zambullirme en eso que antaño se consideraba cultura hacker por mi cuenta, de manera autodidacta. Siempre fui un crío que le encantaba trastear con la tecnología y toquetear los aparatos electrónicos que había por casa.

Al llegar a la universidad, me encontré que hasta el 3er curso no podría tocar el que había sido mi hobby durante años. Tras haber pasado un año replanteándome si había elegido bien el grado que quería estudiar, decidí pasar a la acción y buscar gente de mis mismos afánes y gustos.

Por supuesto, la curiosidad por las comunicaciones inalámbricas vía WiFi fue en ese primer año, a través de mi compañero Carlos.

En mi segundo año de carrera di mis primeras ponencias de ciberseguridad, a través de la II Semana de la Informática en un ciclo de ciberseguridad.

Mi recorrido por la ciberseguridad volvió a crecer de manera exponencial, pero no estaba solo. En mi camino se unieron mis compañeros. Compañeros con los que he compartido mil y una aventuras a través de la asociación LibreLabUCM. Si de

algo estoy orgulloso y ha de quedar por escrito, es de haber compartido mis años de universidad con ellos.

Todo ese conocimiento, todo lo aprendido por el camino en mis años de universidad, unido al temario del propio grado de Ingeniería Informática es lo que nos ha llevado hasta aquí. WiFi-Pocket refleja todo ese conocimiento recogido. Refleja toda esa curiosidad catalizada en una sola herramienta de hacking.

1.4. Estructura del documento

Para poder realizar el presente Trabajo de Fin de Grado era importante tener una amplia visión del estado actual de la ciberseguridad. La investigación presentada en el capítulo 2 se focalizará en las comunicaciones inalámbricas vía WiFi, los ataques que pueden darse en esta, qué mecanismos existen como medio de detección de ataques y qué herramientas hay y son utilizadas actualmente.

Una vez estudiado y fundamentados todos los aspectos que rodean a la herramienta WiFi-Pocket, en el capítulo 3 se describen los engranajes que componen la aplicación. Podremos ver la subdivisión en herramientas más pequeñas que conforman WiFi-Pocket, la explicación de cómo están ordenados nuestros ficheros en el repositorio y cómo se ha gestionado la información en las bases de datos.

Tras explicar la arquitectura de la aplicación, en el capítulo 4 se describe la tecnología que se ha utilizado para su implementación, justificando por qué ha sido elegida.

Una vez explicada toda la tecnología, en el capítulo 5 se detalla sobre que dispositivos se puede y se ha implementado el desarrollo y uso de WiFi-Pocket.

El capítulo 6 aborda la metodología empleada en el proyecto actual, y detalla cómo ha sido el proceso de desarrollo de este proyecto desde su inicio descubriendo todos los cambios y adaptaciones que se han realizado.

En el capítulo 7 se exponen los resultados obtenidos. Por un lado, se muestran los datos estadísticos de la aplicación y, por otro, se muestra la presentación del proyecto, a través de la divulgación mediante ponencias, que se ha realizado.

Por último, en el capítulo 8, se presentan las conclusiones y el trabajo futuro necesario para ampliar la funcionalidad de la herramienta. Como se verá WiFi-Pocket es una herramienta que se ha construido de forma modular, para que se pueda adaptar a los futuros vectores de ataque que vayan surgiendo, nuevas defensas que se vayan necesitando o para permitir su modificación en cuanto apariencia y estructura.

1.5. Gestión del proyecto

Para el desarrollo de este proyecto se han utilizado las siguientes herramientas:

1.5.1. Implementación

En cuanto a desarrollo de código y escritura de la memoria, se han utilizado los editores de texto libres Visual Studio Code y Emacs. Estos editores destacan por adaptarse muy bien a la escritura de código de diferentes lenguajes de programación gracias a su variedad de módulos.

Nos permiten, por ejemplo, en el caso de LaTeX, editar los archivos, guardar las modificaciones y, automáticamente, se compilará el código y podremos ver dichas modificaciones plasmadas en la memoria en pdf.

1.5.2. Gestión y control de cambios

Para el control de versiones y tareas, se ha utilizado git. Todo el código ha sido alojado en un repositorio de la plataforma GitHub.

La gestión de las tareas y corrección de errores se ha llevado a cabo a través de las Issues del repositorio y se han ido planificando objetivos a través de la sección "Proyectos".

Para los flujos de trabajo, se han implementado las siguientes ramas:

- master: rama principal que contiene las versiones estables de la herramienta.
- develop: rama creada solo para el desarrollo de la herramienta. Supondrá la siguiente versión de la misma y, una vez se obtenga una versión estable, se hará un merge con master.
- memory: rama creada solo para el desarrollo de la memoria del Trabajo de Fin de Grado.

Se planteó la creación de nuevas ramas denominadas features para el desarrollo de nuevas características de la aplicación que partirían de la rama develop, pero al ser una única persona la que modificaría el estado de las ramas, esta opción se desestimó.

Chapter 1

Introduction

"The original idea of the web was that it should be a collaborative space where you can communicate through sharing information."
— Tim Berners-Lee

1.1. Goals

The main objective of this project is to provide both Blue Team and Red Team teams with the necessary tools to facilitate their work.

As a secondary objective, it is intended that any user can make use of the project regardless of their knowledge in order to protect their communications infrastructure.

Another of the objectives that would be met in parallel to those mentioned above is the task of keeping the population safe from external attacks, which may be detrimental to society.

WiFi-Pocket consists of a suite of tools oriented to WiFi technology using or not using probes and a control panel. All this in an easy and simplified way, oriented to the use of any user.

1.2. Motivation

Currently, society tends towards the use of wireless networks of WiFi type. These networks allow the user to connect to the Internet without the need to make a physical connection, by cables.

Access to these networks is done through an access point. Although this access point may require a password to navigate through it, it may also be open. These passwordless access points commonly belong to shops or stores that require the user's private data if they want to connect to them.

The exchange of information between a user and an access point is made publicly, that is, the information is sent in the form of packets through waves where anyone can capture them. This capture does not imply that the end user runs out of the package, it simply takes place to read it by an external user. The mere fact

of intercepting packets does not require any connection to the access point, and does not have a record indicating that such activity is being carried out. It is an act without repercussion that cannot be detected.

Protected networks add an encryption layer to the packets preventing the content of the packet from being known or what information it is carrying. If a person wanted to extract such data, they would need to have access to the security measures implemented by the access point.

An attacker could exploit vulnerabilities present in the encryption layers applied to packets sent between devices connected to a router or make use of social engineering tricks in order to attack users of a network. These users may not be aware that they are being attacked or they may have already been attacked and not noticed. The purpose of these attacks tends to be to steal personal information or gain full access to users' devices.

These attacks can be applied not only to individual users but also to large infrastructures. This is because, at present, there is no infrastructure that does not depend on the implementation of a device via WiFi for its users and / or employees.

It is proposed in this Final Degree Project the creation of a tool that allows to emulate being an attacker at the same time that allows to detect in real time attacks that are carried out on a wireless communications infrastructure via WiFi. To create this tool, you will need to dump the software on several hardware devices that will be called probes connected to a central web panel. The information they exchange will be made on their own virtual network, preventing anyone from accessing the information.

1.3. Personal motivation

The curiosity about cybersecurity began in my life the day that caught my attention at age 12, being 1 of ESO. I started to dive into what was once considered a hacker culture by my own, in a self-taught way. I was always a kid who loved to mess with technology and play with the electronic devices that i had in my house.

Upon arriving at the university, I found that until the 3rd year I could not have my hobby. After having spent a year rethinking if I had chosen well the degree I wanted to study, I decided to take action and look for people of my same desire and tastes.

Of course, the curiosity about wireless communications via WiFi was in that first year, through my partner Carlos.

In my second year of my career, I gave my first cybersecurity presentations, through the "II Semana de la Informática".

My journey through cybersecurity grew exponentially, but I was not alone. On my way my teammates joined. Partners with whom I have shared a thousand and one adventures through the LibreLabUCM association. If I am proud of something and it must be written, it is to have shared my university years with them.

All that knowledge, everything learned along by the way in my university years, with the syllabus of the degree of Computer Engineering is what has led us here. WiFi-Pocket reflects all that knowledge collected. It reflects all that curiosity catalyzed in a single hacking tool.

1.4. Document structure

In order to carry out this Final Degree Project it was important to have a broad vision of the current state of cybersecurity. The research presented in the chapter 2 will focus on wireless communications via WiFi, the attacks that can occur in this, what mechanisms exist as a means of detecting attacks and what tools are and are currently used.

Once all aspects surrounding the WiFi-Pocket tool have been studied and based, the gears that make up the application are described in the chapter 3. We can see the subdivision into smaller tools that make up WiFi-Pocket, the explanation of how our files are sorted in the repository and how the information in the databases has been managed.

After explaining the architecture of the application, the 4 chapter describes the technology that has been used for its implementation, justifying why it has been chosen.

Once all the technology has been explained, the chapter **??** details what devices you can use and has implemented and developed the use of WiFi-Pocket.

The chapter 6 deals with the methodology used in the current project, and details how the development process of this project has been since its inception, discovering all the changes and adaptations that have been made.

In the chapter 7 the results obtained are presented. On the one hand, the statistical data of the application are shown and, on the other, the presentation of the project is shown, through the dissemination through presentations, which has been made.

Finally, in the chapter , the conclusions and the future work necessary to extend the functionality of the tool are presented. As you will see, WiFi-Pocket is a tool that has been built in a modular way, so that it can adapt to future attack vectors that may arise, new defenses that may be needed or to allow its modification in appearance and structure.

1.5. Project management

The following tools have been used for the development of the project:

1.5.1. Implementation

The free text editors Visual Studio Code and Emacs have been used for code development and memory writing. These editors stand out for adapting very well to the code writing of different programming languages thanks to their variety of modules.

They allow us, for example, in the case of using LaTeX, editing the files, saving the modifications and, automatically, the code will be compiled and we will be able to see live those modifications embodied in the pdf memory.

1.5.2. Management and version control

For version and task control, git has been used. All code has been hosted in a repository of the GitHub platform.

The task management and bug fixing has been carried out through the Issues of the repository and have been translated into objective projects through the "Projects" section.

For workflows, the following branches have been implemented:

- master: main branch that contains stable versions of the tool.
- develop: branch created only for tool development. It will be the next version of the stable version in master branch.
- memory: branch created only for the development of the memory of the Final Degree Project.

The creation of new branches called features was proposed for the development of new features of the application that would start from the develop branch. This option was rejected because just a single person who can modify the state of the branches.

Capítulo 2

Estado del Arte

"Internet es la primera cosa que la humanidad ha construido y que la humanidad no entiende. Es el experimento más grande de anarquía que hemos tenido"
— Eric Schmidt

2.1. Situación de la tecnología

El mundo está evolucionado hacía un paradigma digital como nunca antes se ha visto. Donde antes se acudía a una biblioteca para consultar los pesados libros que llenarían de ideas el intelecto de los estudiantes ahora simplemente uno tiene que conectarse, teclear su búsqueda deseada y la encontrará, sin moverse del sitio, en segundos. No tenemos mayor contacto que el tacto de unas teclas, la ligereza de un ratón y el venenoso y desolador brillo de las pantallas de nuestros dispositivos.

Echando la vista atrás uno puede contemplar como los líquidos sueños y aspiraciones que impregnaban las películas de ficción están pasado al estado sólido de la realidad. El hombre ya ha llegado a la luna, vive en el espacio, puede respirar bajo el agua, puede ver de noche, puede construir edificios que atraviesan el cielo.

Sin embargo... No todo son ventajas, la dependencia que ha emergido en la raza humana hacia la tecnología ya ha empezado, asentandose en los pilares de la sociedad y creciendo de manera exponencial. Nuestras futuras generaciones nacen llorando y portando un dispositivo tecnológico con ellos. La gente visualiza su vida y la vida de quienes les rodean a través de una pantalla sin alma, sin sentimientos, sin emociones. El problema radica en la necesidad humana de posesión de esa pantalla para poder satisfacer su condición como ser sociable. Hay una presión social y evolutiva que nos lleva a olvidar lo que las generaciones anteriores aprendieron, lo que las generaciones anteriores consideraban como felicidad, lo que las generaciones anteriores consideraban como vivir la vida.

El ser humano siempre ha estado dotado del don de la comunicación. Se dice que creamos sociedad cuando creamos lenguaje. Hoy en día las comunicaciones se han expandido. No hay barreras, no hay límites. De no poder comunicarnos hemos pasado a tener que enviar señales de humo, a ir a caballo con un mensaje que

podían tardar casi un mes en llegar, a escribir en un papel nuestras más pensadas y calculadas palabras que bien pudieran ser transportadas vía aérea, para finalmente ni siquiera necesitar nada de eso. Ya no hace falta escribir una carta de papel, ya no hace falta ir a ninguna cabina telefónica para poder establecer contacto con otra persona. Ya ni tan siquiera hace falta tener que conectarse vía cable a la red y teclear nuestro correo electrónico.

Las ondas de microondas han sustituido al cable físico que nos unía a la red. Bares, restaurantes, centros comerciales, la casa de nuestra abuela que vive en un pueblo alejado de la mano del señor,... Todos ellos han sido arrastrados por la ola tecnológica depositando en sus hogares un dispositivo que, vía WiFi, nos permite conectarnos a la red.

Cada vez son más los dispositivos que vienen con opción para conectarse a internet. Hacernos el plato de cocido gallego de nuestra madre está a un solo clic. Encargar una impresora 3D y que nos llegue al día siguiente esta a una sola pulsación en el touchpad. Podemos tenerlo todo realizando un fácil gesto, desembolsando una gran cantidad de dinero y al momento.

2.2. Ciberseguridad

Siguiendo el hilo de la sección anterior, no todo es tan bonito y deslumbrante como aparenta ser. Pese a habernos comportado como dioses creando un mundo paralelo al nuestro, al real, eso no significa que ese mundo sea de color de rosas.

Al igual que la tecnología está creciendo también lo hace la inseguridad digital. La masiva producción de nuevos hallazgos tecnológicos para satisfacer el voraz y caprichoso apetito humano hace que estos deban ser sacados en el menor tiempo posible y sin prestar atención a la seguridad.

En la tecnología vienen implícitas la comodidad y la conversión de lo complejo a sencillo. La tecnología implica progreso. Las centrales nucleares están conectadas a una red, los hospitales guardan historiales clínicos en discos duros alojados en sus servidores, los barcos envían su ubicación geográfica por internet, la gente guarda su información personal en la nube. ¡Las casas ahora ya casi funcionan solas!

Pero, por mucho que nos aporte la tecnología, el ser humano está abocado al fracaso si no se predispone a afrontar los desafíos de la inseguridad digital.

En el año 2004, en España, surgió el Centro Criptológico Nacional, como un centro independiente adscrito al Centro Nacional de Ciberseguridad, al que se le asignaron las funciones relativas a la seguridad de las Tecnologías de la Información y la protección de la información clasificada.

En el año 2006, la Organización del Tratado del Atlántico Norte (OTAN) incluyó Internet como uno de los dominios de guerra. Así pues, quedan como dominios el mar, la tierra, el aire, el espacio y el ciberespacio. Palabras textuales del secretario general de la OTAN, Jens Stoltenberg: "Un ciberataque puede desencadenar una acción colectiva de defensa, porque contemplamos los ciberataques como algo que pueden causar mucho daño y ser muy peligroso".

En el 2014, surgió el Instituto Nacional de Ciberseguridad, INCIBE. Entre sus funciones destaca el desarrollo de la ciberseguridad y la confianza digital de ciudadanos y empresas, sobre todo de aquellas que gestionan infraestructuras públicas.

Cada vez son más las infraestructuras creadas en apoyo a la ciberseguridad para solventar el problema de la seguridad digital. Esto es solo el inicio del recorrido hacia

un mundo conectado y seguro. El camino empieza ahora.

2.3. Red Team y Blue Team

Con el paso del tiempo los términos de Red Team y Blue Team van cobrando más relevancia, llegando a ofrecer puestos de trabajo específicos para cada grupo, y no uno solo que englobe ciberseguridad.

Pero, ¿a qué hacen referencia? Hacemos uso de los términos Red Team y Blue Team para referirnos a los equipos encargados de simular ser los atacantes o ser las personas encargadas de la defensa de la infraestructura de una organización.

2.3.1. Red Team

La misión del Red Team consiste en emular las tácticas, técnicas y procedimientos que suelen realizar los ciberdelincuentes. El objetivo es conseguir datos concretos sobre cómo responderá una empresa a ataques en el mundo real, encontrar brechas de seguridad, identificar deficiencias de seguridad en el personal y finalmente mejorar la situación de la empresa. Previamente la empresa cliente debe definir claramente el objetivo de la campaña, bajo un contrato de confidencialidad, indicando también el alcance sobre el que se realizará el ataque. En este contrato es donde se define hasta dónde se puede llegar.

El Red Team no es tán metódico como las pruebas de pentesting. La emulación de un ataque, al tratarse de un entorno real, hace que cada prueba difiera significativamente. Algunas campañas pueden tener el foco en adquirir información personal, credenciales o tarjetas, mientras otras pueden tener como objetivo obtener el control de administrador del dominio.

Para las campañas de Red Team, lo normal es ignorar completamente el controlador de dominio. Esto es debido a que actualmente la mayoría de las compañías lo protegen activamente. Esta es una de las diferencias entre el Red Team y los pentesters, y es que los pentesters suelen tratar de escalar privilegios para llegar a tener acceso al controlador del dominio. Esta protección activa, suele implicar conjuntos de aplicaciones en la lista blanca, monitorización de integridad, reglas de IDS / IPS / HIPS e incluso algunas más. Ya que el Red Team no debe ser descubierto, ha de permanecer discreto. Es por ello que hay que evitar en todo lo posible ejecutar análisis de vulnerabilidades de la red interna. ¿Por qué? Los escaneos de vulnerabilidades hacen muchísimo ruido en la red y a día de hoy son bastante fáciles de detectar, por lo que estaríamos corriendo un riesgo muy alto de ser descubiertos.

Las campañas de Red Team suelen llevarse a cabo en largos periodos, que pueden durar varios meses. Esto se debe a que necesitamos realizar ataques reales, no sólo con técnicas usadas en test de intrusión convencional sino también aplicando ingeniería social. El resultado final, no se parecerá a la lista de vulnerabilidades que encontraríamos en pruebas de pentesting, los hallazgos del Red Team deben orientarse más hacia las brechas en los procesos, políticas, herramientas y habilidades del Blue Team. En el informe final, es posible que aparezcan algunas de las vulnerabilidades encontradas, pero la mayoría de los hallazgos serán brechas en la seguridad de la empresa.

La información que devuelve el Red Team debe ser de valor. No se trata de la cantidad de recuentos de vulnerabilidades totales o de la criticidad de las vulnera-

bilidades individuales; se trata de probar cómo están funcionando los mecanismos de seguridad de la empresa. Dos de las métricas que se usan en estas campañas son el tiempo para detectar y el tiempo para mitigar. Estos no son conceptos nuevos, pero son valiosos para el Red Team.

El tiempo de detección, es el tiempo entre la aparición inicial del incidente y cuando un analista detecta y comienza a trabajar en el incidente. Digamos que un usuario recibe un correo electrónico que contiene un malware que el usuario ejecuta en su sistema. A pesar de que su antivirus, sistema de seguridad basado en host o herramientas de monitorización pueden activarse, el tiempo registrado es cuando el analista crea ese primer ticket.

El tiempo para mitigar es la métrica secundaria a registrar. Esta línea de tiempo queda reflejada cuando se produce el bloqueo del firewall, el DNS skinhole (sumidero de DNS) o el aislamiento de red. La otra información valiosa para registrar es cómo los equipos de seguridad trabajan con TI, cómo la administración maneja un incidente crítico y si los empleados entran en pánico. Con todos estos datos, podemos construir números reales sobre cuánto está en riesgo una empresa, o la probabilidad de que se vea comprometida.

2.3.2. Blue Team

El Blue Team está compuesto por un grupo de expertos especialistas en ciberseguridad que monitorizan, rastrean, controlan y realizan análisis de seguridad de los sistemas de una empresa.

Podemos encontrar similitudes entre RedTeam y BlueTeam. Mientras que el RedTeam se encarga de emular ser un atacante mediante tácticas y técnicas concretas, el BlueTeam se encarga de buscar formas de defender, pudiendo llegar incluso a modificar los mecanismos de defensa evitando así que la respuesta a un incidente sea mucho más grave.

Ambos equipos necesitan conocer las mismas tácticas, técnicas y procedimientos maliciosos por los cuales un atacante podría ir directo a por un activo con el objetivo de construir estrategias de respuesta.

La actividad del BlueTeam no consiste solo en conocer los posibles ataques que se puedan recibir. Constantemente deben establecer medidas que fortalezcan toda la infraestructura de activos. Para ello, se usa software como el caso de los IDS (sistema de detección de intrusos). Este software, mediante reglas previamente redactadas, proporciona un análisis continuo de actividades inusuales y sospechosas. En base a este análisis se pueden tomar ciertas medidas tanto a tiempo real como a posteriori.

La mejor forma de detectar posibles intrusiones es observando el tráfico de datos, el comportamiento que tienen los activos así como quién y a dónde se realizan las conexiones por un usuario de forma habitual.

A través de esta observación los equipos de BlueTeam son capaces de detectar los incidentes en el menor tiempo posible con el fin de impedir por ejemplo el robo o perdida de información sensible. Esta perdida podría provocar graves consecuencias económicas para una empresa, dañar su reputación o incluso ver muy gravemente perjudicada su imagen.

Cabe destacar que en el proceso de defensa se implementan algunos señuelos con el objetivo de poder avistar nuevos futuros atacantes pudiendo estudiar cómo

se comportan, cuáles son sus tácticas y permitiendo recabar información acerca de la procedencia del ataque.

Estos equipos deben ser conscientes de que en cualquier momento pueden sufrir un ciberataque y que la mejor defensa que tendrán será su preparación. Deben estar preparados para reaccionar, minimizar los daños e intentar que el servicio no se vea apenas perjudicado.

2.4. WiFi

Uno de los problemas que poseen las redes Wifi, es que sus datos están "en el aire" y van en claro. Cualquier usuario puede esnifar el tráfico para detectar que es lo que acontece en las diferentes redes a su alrededor. En cambio, en una red cableada, necesitamos que un usuario para poder leer todo el tráfico este conectado a nuestro cable.

2.4.1. ¿Qué son las Redes Wi-Fi?

Debemos diferenciar lo que es el estándar 802.11 y la marca Wi-Fi. El término Wi-Fi viene dado por la marca registrada por Wi-Fi Alliance, que es una organización sin ánimo de lucro que promueve la tecnología Wi-Fi y certifica aquellos productos que cumplen con los estándares 802.11. Los productos que cumplen con la certificación, se les permite ulizar el logotipo de Wi-Fi.

2.4.2. Estándar 802.11

Es el estándar IEEE para redes inalámbricas WLAN. Desarrollado durante los años noventa, especifica las normas de funcionamiento de una red de área local inalámbrica. Este estándar actualmente se compone de al menos 23 protocolos. Algunos de los ejemplos son: 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac.

2.4.3. Protocolos de red

Dentro de estos protocolos nos podemos encontrar con ARP, UDP, OLSR, ICMP/PING, TCP, Meshz, IP.

2.4.4. Capas de Internet

La arquitectura de Internet y, concretamente el protocolo TCP/IP (ver figura 2.1), se suele implementar en un modelo de 5 capas, de las cuales distinguimos la capa de aplicación, la capa de transporte, la capa de red, la capa de enlace y la capa física. Por sus definiciones:

- Capa de aplicación: realiza el intercambio de mensajes entre dos programas (aplicaciones). Su comunicación es de extremo-extremo con la lógica de la aplicación. Los protocolos que circulan por ella son: HTTP, SMTP, FTP, TELNET, DNS, ...

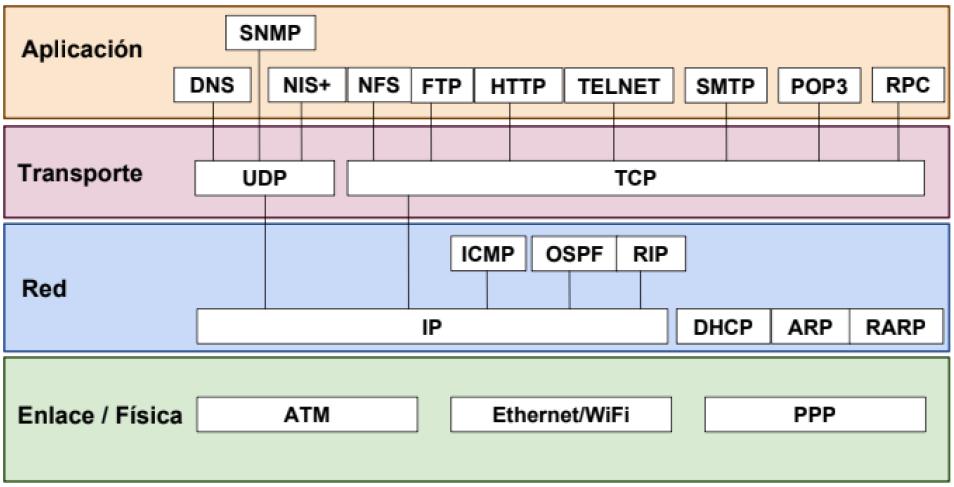


Figura 2.1: Arquitectura de protocolos TCP/IP

- Capa de transporte: realiza una comunicación de extremo-extremo. Encapsula los mensajes de la capa de aplicación en un segmento o datagrama. Envía un mensaje de una aplicación y lo entrega a la aplicación correspondiente en el otro extremo. Se distinguen dos protocolos principales:
  - TCP, protocolo de transporte orientado a conexión: control de flujo, errores y congestión
  - UDP, sin conexión (mensajes independientes). Simple, sin las ventajas anteriores.
- Capa de red: es la responsable de la comunicación entre los hosts y de enviar los paquetes por el mejor camino posible. Se asocia con los protocolos de internet:
  - Define el formato del paquete (datagrama)
  - La forma en que se designan los hosts (direcciones)
  - Encaminamiento (unicast y multicast)
  - Protocolos asociados: IGMP, ARP, ICMP, DHCP
  - No ofrece control de errores, congestión o flujo
- Capa de enlace: permite la transmisión de los datagramas por el enlace. El datagrama se encapsula en un marco (frame). No se especifica un protocolo en particular. Pueden ofrecer corrección/detección de errores.
- Capa física: Responsable del envío de bits por el enlace en particular. Realiza la codificación, conversiones (digital-digital, digital-analógica, ...), multiplexación, ... La comunicación sigue siendo lógica. Medio de transmisión, envío efectivo de la información como señales electromagnéticas.

2.4.5. Paquetes de datos

Entendemos por paquete de datos o paquete de red como aquellos bloques en los cuales se divide la información que se enviará en el nivel de red.

Como veremos más adelante en la memoria en los programas de escaneo de paquetes de red (con el objetivo de esnifar tráfico), podemos distinguir los siguientes tipos: Management, Control y Data. A su vez, estos tipos se dividen en subtipos. Para tener una visión gráfica de ello, se presenta en la tabla 2.1 que muestra la división de tipos y su subdivisión en subtipos:

Management	Control	Data
ASOCRQ	BEAMRP	DATA
ASOCRP	VHTNDP	DCFACK
REASRQ	CTWRAP	DCFPLL
REASRP	BACKRQ	DCFKPL
PROBRQ	BACK	NULL
PROBRP	PSPOLL	CFACK
TIMING	RTS	CFPOLL
BEACON	CTS	CFCKPL
ATIM	ACK	QDATA
DISASC	CFEND	QDCFCK
AUTH	CFENDK	QDCFPL
DEAUTH		QDCFKP
ACTION		QDNULL
ACTNOA		QCFPLL
		QCFKPL

Tabla 2.1: Tipos de paquetes de red

Lo que a nosotros nos interesará serán los paquetes usados para realizar ataques o identificación de objetivos.

2.4.5.1. Probe Request

Probe request es un tipo de paquete del protocolo 802.11 de WIFI. Este paquete permite al usuario conectarse de manera automática con los puntos de acceso (AP) que el usuario tiene guardados por haberse conectado con anterioridad. Cuando un usuario con su teléfono móvil u ordenador tiene la red WiFi encendida, sin estar conectada a ninguna red, esta permitiendo que su teléfono móvil este constantemente preguntando a los puntos de acceso de su alrededor “eres la red XXX” siendo XXX una red a la cual nos hayamos conectado anteriormente.

Un atacante puede crear un punto de acceso con el nombre del AP al cual podemos conectarnos y engañar a nuestro dispositivo para que se conecte a un Fake AP.

2.4.5.2. Deauth y Auth

Deauth y Auth son un tipo de paquete del protocolo 802.11 de WiFi que envían señales de desconexión del punto de acceso o de conexión con respecto al punto de acceso. Cuando estamos conectados a una red, y posteriormente nos desconectamos, se envía un mensaje de Deauth.

Un atacante puede enviar numerosos paquetes Deauth provocando la desconexión de los dispositivos en una red. Estos paquetes son enviados desde el ordenador del atacante al dispositivo de la víctima. Auth por otro lado es un paquete de autenticación.

Resulta de mucha utilidad, por ejemplo cuando nos quedamos sin poder conectarnos al Wifi de bibliotecas como la María Zambrano, desautenticar a todos los usuarios para poder permitir así a nuestro dispositivo conectarse a la red. Por supuesto, no se recomienda esta práctica.

2.4.5.3. Beacons

Los Beacon frames son un tipo de paquete del protocolo 802.11 de Wifi que contienen toda la información sobre la red inalámbrica, como por ejemplo en qué canal se encuentran, qué tipo de cifrado tiene configurado, cómo se llama la red, etc. Esta información es transmitida en claro, sin ningún tipo de cifrado. Los beacons son transmitidos por los puntos de acceso periódicamente para anunciar la presencia de su red WLAN. Un Beacon frame está formado por una cabecera MACaddress, un cuerpo y un FCS.

A través de estos beacons podemos descubrir los AP que existen a nuestro alrededor.

El envío de estos paquetes puede ser utilizado por un atacante de mil maneras, como por ejemplo enviar beacons de múltiples puntos de acceso falsos en diferentes canales con el mismo nombre dificultando la conexión con el punto de acceso correcto provocando que el usuario tenga difícil acceso a una conexión a internet.

2.4.6. Tarjetas de red

Una tarjeta de red es un componente de hardware que nos permite poder acceder a una red desde nuestro dispositivo.

2.4.6.1. Dirección MAC

La dirección MAC es una identificador único a nivel mundial de cada dispositivo. Está formado por 48 bits (6 bloques de dos caracteres hexadecimales (4 bits)) que permite identificar la totalidad de dispositivos de red tales como las tarjetas de red.

Normalmente, los fabricantes una vez tienen montado el hardware, véase una tarjeta de red, graban en formato binario en una memoria ROM del dispositivo la dirección MAC. Dicha memoria es de solo lectura, lo cual imposibilita poder modificarla directamente en la ROM.

Según lo anterior no podemos modificarla. Cabe destacar que lo que no podemos modificar es directamente la MAC que tenemos almacenada en la ROM. Entonces... ¿qué podemos modificar? Al arrancar el sistema se carga una copia de la dirección MAC en nuestra memoria RAM. En este caso, lo que podemos hacer es modificar

la dirección MAC que está almacenada en la RAM. Dicha dirección será la usada en todas las acciones donde se precise su uso.

Anteriormente habíamos mencionado que la dirección MAC es un identificador único a nivel mundial de cada dispositivo. Existe una entidad, el Instituto de Ingeniería Eléctrica y Electrónica (IEEE), que regula el formato de estos identificadores.. Dicha entidad es el

Una dirección MAC tiene el siguiente formato: "40:AF:E5:F5:FE:50".

Debemos centrarnos principalmente en dos partes:

- Por un lado tenemos la primera parte, el lado izquierdo, el cual consta de 3 bloques. Dicho lado identifica quién es el fabricante del hardware. Esto quiere decir que si en nuestra red vemos una dirección MAC que empieza por "40:AF:E5" esta pertenecerá a Sawyer Technologies.
- La segunda parte, la de la derecha, consta también de 3 bloques. Esta parte del código es el número de serie que identifica el dispositivo fabricado. El fabricante puede usar el número de serie que quiera pero con la condición de que únicamente puede usar el mismo número de serie una vez. Es decir, los 3 últimos bloques deben ser únicos para un único dispositivo. Si se fabricase otro, no se puede usar la terminación "F5:FE:50" porque ya ha sido usada.

2.4.6.2. Modo monitor

Se conoce el modo monitor, también llamado modo promiscuo o modo escucha, como el modo que nos permite capturar todo el tráfico que circula por una red. Dentro de este modo, buscamos poder capturar todos los paquetes que circulan a través de Wifi de las redes que se encuentran a nuestro alrededor y que son alcanzables por nuestra tarjeta de red.

No todas las tarjetas de red admiten modo monitor. Es por ello que deberemos tener cuidado en el uso que hagamos de nuestras tarjetas de red. Para nuestro caso, las tarjetas de red comentadas en el apartado Hardware de la memoria sí disponen de este modo.

Para poder establecer el modo monitor hay diferentes formas. La primera es la usada comunmente por pentesters, los cuales suelen hacer uso de la herramienta airmon-ng. Esta herramienta viene por defecto en las distribuciones orientadas a realizar pruebas de auditoría wifi. El comando para establecer el modo monitor a través de airmon-ng es el siguiente:

```
1    sudo airmon-ng start <interfaz>
```

La segunda opción, más recomendable y mayoritariamente usada, es iwconfig. Esta herramienta se encuentra obsoleta, por lo cual usaremos iw. Ambas se encuentran en los repositorios oficiales de las distribuciones GNU/Linux más conocidas como Debian o Arch.

En este caso, para establecer el modo monitor a través de iw, necesitamos ejecutar:

```
1    sudo ip link set <interfaz> down
2    sudo iw <interfaz> set monitor control
3    sudo ip link set <interfaz> up
```

Como para nuestro proyecto buscamos ahorrar el mayor espacio de memoria posible y consumir el menor tiempo de procesamiento posible, utilizaremos la segunda opción.

2.4.7. Ataques WiFi

Actualmente existen multitud de ataques hacia la infraestructura WiFi e incluso hacia los usuarios que hacen uso de este método de comunicación inalámbrica. Ya no solo ataques, sino multitud de herramientas que producen estos ataques. Para el presente TFG se han estudiado los diferentes ataques y modos de evasión de los protocolos de seguridad, tales como "Ataque Chop Chop", "Denegación de Servicio", "Ataque por fuerza bruta (Diccionario o PINs hacia WPS)", ...

A continuación, se van a describir los ataques que se han estudiado haciendo uso de libros como Ángel Ramos Varón et al. (2014)

2.4.7.1. Ataque por desautenticación

Este ataque se conoce bajo el nombre de DeauthFlood. Estos ataques normalmente sirven como paso previo a ataques más complejos.

El ataque puede realizarse sobre el cliente, o sobre el punto de acceso. Si lo realizamos sobre el cliente, estaremos desautenticando al usuario que estaba conectado a la red impidiendo que pueda realizar una navegación normal. En caso de ser realizado sobre el AP, el cliente seguirá pensando que está conectado, pero los datos solicitados al AP estarán bloqueados, por lo que su navegación estará bloqueada como se puede ver en la figura 2.2.

Sin embargo, hay razones por las que este ataque puede fallar:

- Es posible que el atacante se encuentre demasiado lejos del cliente al que quiere desautenticar. Para comprobarlo será necesario verificar que llega un ack indicando que se ha recibido el paquete.
- Es posible que el modo de la tarjeta del atacante no sea compatible con el modo de red que tiene el cliente. Algunos de estos modos se han visto en la memoria en el apartado 2.4.2 y corresponderian a b, g, n. Si están en diferentes modos, el cliente no verá el paquete.
- Algunos puntos de acceso no aceptan paquetes deauth de estilo broadcast. Para poder desautenticar usuarios en este caso, lo mejor será ir desautenticando a clientes uno por uno, de forma individual.
- Puede que los clientes se estén reconectando demasiado rápido de lo que se están desconectando. Se pueden calcular los tiempos de reconexión esnifando el tráfico.

El modo de detección de este ataque consiste en detectar un número de paquetes con el flag DEAUTH activo enviados en una franja de tiempo concreta.

2.4.7.2. Ataque por autenticación

Este ataque se asemeja al ataque basado en desautenticación por las consecuencias que causa sobre el usuario. Un atacante puede inundar un punto de acceso con

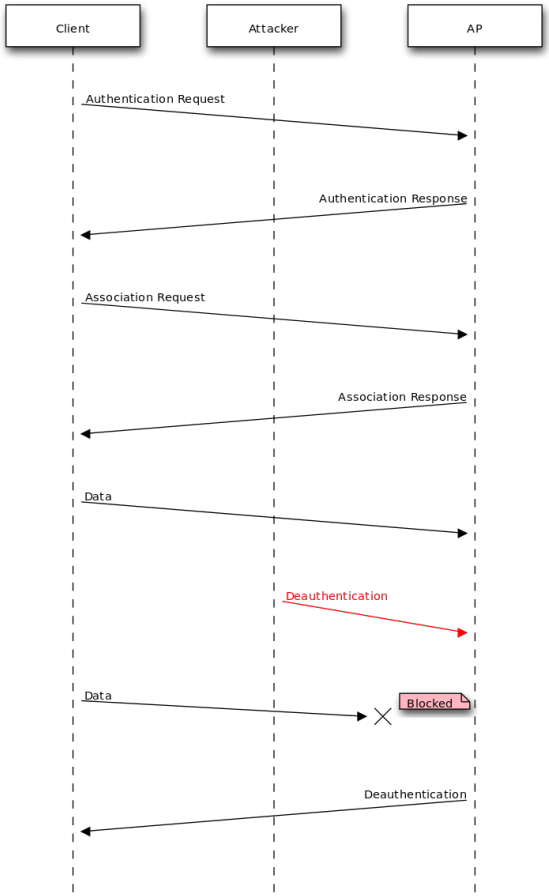


Figura 2.2: Desautenticando a través del AP

peticiones de autenticación usando direcciones MAC falsas, impidiendo la conexión de un usuario legítimo.

Este ataque tiende a considerarse un ataque DoS (Denial of Service - Denegación de Servicio) por el envío masivo de peticiones al punto de acceso WiFi.

El modo de detección de este ataque consiste en detectar un número de paquetes con el flag AUTH activo enviados en una franja de tiempo concreta.

Podemos ver un ejemplo visual en la figura 2.3

2.4.7.3. Suplantación de direcciones MAC

Un usuario puede obtener la clave de nuestro punto de acceso y acceder a nuestra red. Para evitar esto, se suelen limitar los accesos configurando "listas blancas" con las direcciones MAC que queramos que estén conectadas a nuestro AP.

Un atacante, en caso de tener la clave y no poder acceder a un punto de acceso, puede esnifar el tráfico en busca de las direcciones MAC de los clientes que haya conectados y cambiar su dirección MAC por una de ellas. Con esta nueva dirección

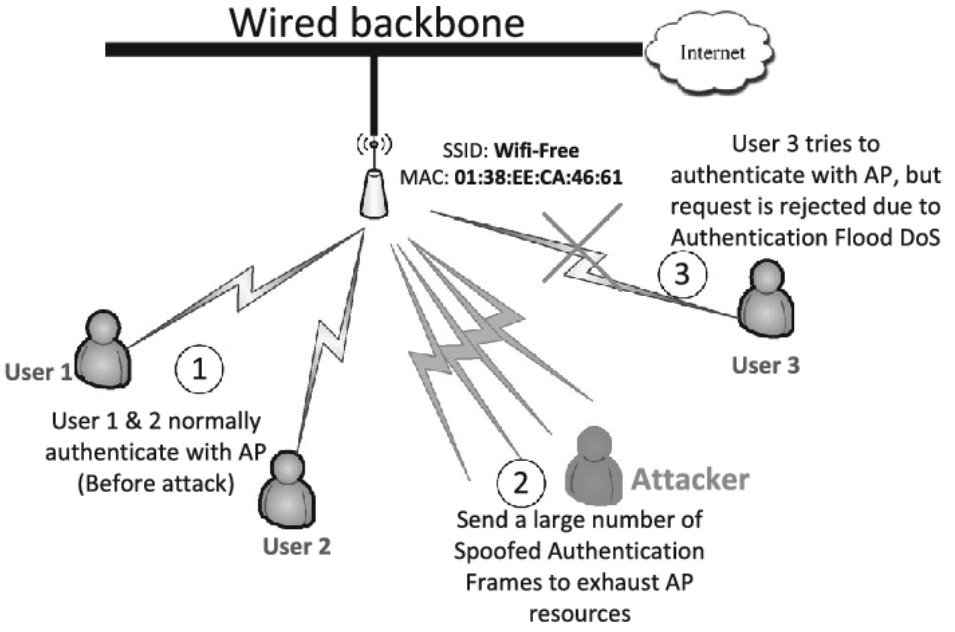


Figura 2.3: Ataque por autenticación

MAC, puede evadir las medidas de seguridad de las listas blancas.

El modo de detectar estos posibles ataques, es controlar direcciones MAC duplicadas o conectadas fuera de horario.

2.4.7.4. Captive Portal

La realización de un Captive Portal viene asociado a la creación de un punto de acceso WiFi. Normalmente los atacantes crean un punto de acceso WiFI gratuito para que los usuarios se conecten a él y así poder interceptar su tráfico. Este tráfico, al menos en la actualidad, irá cifrado bajo protocolos como HTTPS. Pese a ello, aunque pensemos que esto nos adjudica un cierto nivel de seguridad, hay métodos de ataque que nos permiten forzar al usuario a hacer uso de HTTP. El hacer uso de HTTP implica que la navegación del usuario irá sin cifrar, permitiendo a un atacante interceptar posibles credenciales o información sensible.

La creación del punto de acceso puede realizarse desde dos perspectivas. La primera es la creación de un punto de acceso Fake, que sea grauito, y esperar que la sed por internet de los usuarios haga mella y se conecten. La segunda opción es esnifar el tráfico para detectar qué puntos de acceso están buscando los diferentes dispositivos a través de los paquetes PROBREQ. Esta segunda opción nos permitiría en una primera instancia atacar a un solo objetivo, salvo que estemos en una empresa. En caso de estar en una empresa o una universidad, como la Universidad Complutense de Madrid, se crearía un punto de acceso similar al ya existente, se aumentaría la potencia de la señal y se buscaría desautenticar a los usaurios del punto veraz de origen. Esto provocaría que los dispositivos se conectasen a nuestra red disponiendo nosotros de todo el control sobre la misma.

Una vez decidido de qué forma realizar el punto de acceso, queda la realización del Captive Portal. El Captive Portal consiste en la creación de un portal cautivo de acceso. Este portal será lo primero que se encuentre el usuario al conectarse sobre nuestra red. Este portal es recomendable que se asemeje o contenga referencias sobre el propio punto de acceso que queremos emular. Es decir, si queremos emular el punto de acceso de la red UCM, deberemos tener la copia del propio portal.

Una opción que suelen barajar los atacantes, es permitir realizar el inicio de sesión a través de redes sociales. La clave de ello es redirigir las peticiones DNS hacia servidores que contengan las propias web falsas que estarían emulando las oficiales. Esto conlleva que un cliente se conecta a través de la plataforma a un señuelo para que escriba su usuario y contraseña de la red social en cuestión.

Este ataque será el que se incluya como parte ofensiva del trabajo y sobre el que construiremos la parte defensiva.

2.4.7.5. Gemelo Malvado

Conocido en inglés como Evil Twin, este ataque busca crear un punto de acceso igual que otro ya existente. Este nuevo punto de acceso creado por el atacante pretende confundir al usuario para que este se conecte a él.

La forma más común de crear un punto de acceso es tan solo emulando el nombre de la red (ESSID). El cliente no tiene por qué ser el propio usuario que usa un dispositivo, sino que puede ser el dispositivo en sí.

Por un lado, un usuario puede ver una red con un nombre conocido y conectarse a ella.

Por otro lado, nuestro dispositivo puede tener en el historial de conexiones una red a la que previamente se había conectado, y conectarse a una red que, sin ser la misma, contiene el mismo ESSID. Este proceso lo hace de forma automática. Muchos dispositivos poseen reglas para advertir al usuario de que puede ser un punto de acceso suplantado. Estas reglas pueden ser, por ejemplo, la comprobación del ESSID.

Si el dispositivo se conecta de forma automática, el atacante podrá tenerle en su propia red y realizar diversos ataques sobre él.

Se puede ver este ataque a través de la figura 2.4.

2.5. IDS

El término IDS es usado para referirnos a Intrusion Detection System. Su traducción al español es Sistema de Detección de Intrusos. Su función consiste en detectar un uso no autorizado de un activo.

Consideramos como activo un dispositivo personal, un dispositivo o una infraestructura de nuestra organización, o nuestra propia red.

El uso no autorizado puede ser un uso fuera de lo común o bien puede ser un ataque hacia nuestros activos. Esto viene definido en las reglas que aportamos al IDS para su configuración.

Estos IDS pueden estar implementados en los Firewalls.

Cuando queremos referirnos a IDS vía WiFi, nos referimos a ellos bajo el nombre de WIDS. En la actualidad, todos los IDS para detectar intrusiones vía WiFI son

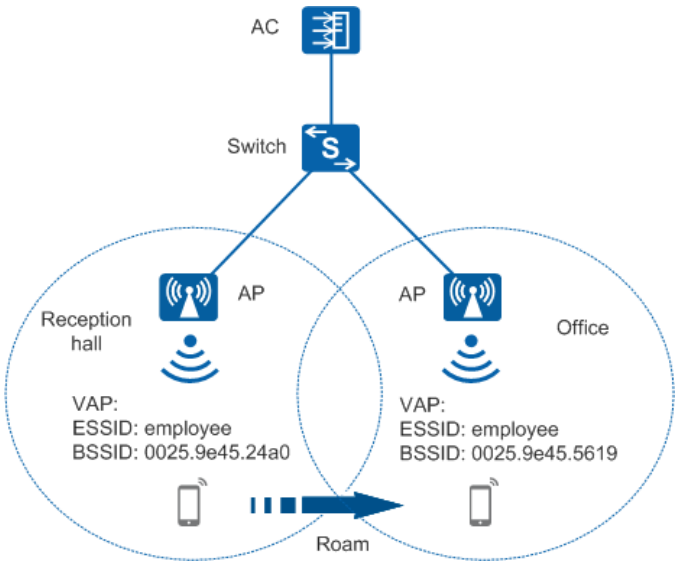


Figura 2.4: Ataque Evil Twin

de pago y no se encuentra una alternativa libre desarrollada y pública.

2.5.1. ¿Cómo funciona?

El funcionamiento de un IDS se basa en la detección de un comportamiento por parte de un usuario o en un tráfico de red fuera de lo normal. Si queremos tener un IDS vía WiFi, la red a analizar será la red inalámbrica que establezcamos. En WiFi los paquetes pueden ir sin cifrar y son visibles incluso por usuarios no conectados a la misma red por lo que podremos tener un IDS que controle las diferentes redes que estén a nuestro alrededor.

Normalmente nos solemos encontrar tres partes bien diferenciadas dentro de la estrucutra de un IDS:

- Sensores: son los encargados de recolectar el tráfico proveniente de una red y de enviarlo al analizador.
- Analizador: se encarga de buscar actividades sospechosas en los datos enviados por los sensores. Esta búsqueda puede realizarse, por ejemplo, en base a unas reglas escritas por el administrador. Un ejemplo de regla sería aquella que detectase a un dispositivo enviando paquetes ICMP a todas las direcciones IP de una red. Otro ejemplo, es realizar una búsqueda en las desviaciones estadísticas de un comportamiento habitual por parte del usuario. En cuanto detecta una actividad que no se considera normal, se envía una alerta al administrador de la interfaz.
- Interfaz de administrador: desde la cual recibimos las alertas y se las mostramos al usuario. Algunos incluso nos permiten a través de esta interfaz configurar las reglas para la generación de alertas.

2.5.2. Tipos de IDS

Dentro de los IDS nos podemos encontrar varios tipos bien diferenciados. Una forma de categorización es en sistemas pasivos y sistemas reactivos.

- Sistemas pasivos: los sistemas pasivos son aquellos que detectan un uso no adecuado de nuestros activos, almacena dicha información y manda una señal de alerta directamente al administrador o que se almacena en la base de datos.
- Sistemas reactivos: a diferencia del sistema pasivo, el sistema reactivo responde a dicha alerta como por ejemplo reprogramando reglas que impidan acceso al usuario que hace un uso no adecuado del activo. Estos sistemas que reaccionan a la detección de un posible ataque, se denominan IPS (Intrusion prevention System).

Por otro lado, otra forma de categorización se basa en el modo en que se recolecta la información y su forma de reacción.

- HIDS: un HIDS (HostIDS) detecta posibles modificaciones que se hayan producido sobre un activo y hace un reporte basado en dichas modificaciones.
- NIDS: un NIDS (NetworkIDS) detecta posibles ataques donde se encuentran los activos. Para hacer posible esta detección se debe poner la interfaz de red en modo monitor.
- WIDS: un WIDS (WirelessIDS) detecta posibles ataques en los diferentes puntos de acceso alrededor de un activo, o donde este se encuentre conectado. Apenas hay conocimiento o desarrollo en torno a los IDS basados en WiFi.

A su vez, estos puede dividirse en dos.

- IDS basado en firmas: llamamos firma a un patrón de comportamiento conocido. Un IDS basado en firmas compara el contenido de los paquetes de red con los patrones de comportamiento en ataques. El problema de este tipo de IDS es que cualquier ataque no identificado previamente se considera como uso normal.
- IDS basado en anomalías: este tipo de IDS también hace una comparación pero en lugar de hacerlo en base a comportamientos, lo hace en base a lo que considera normal en una red o en un host. Por ejemplo, si normalmente en una red solo se utilizan los puertos 80, 443 y 22, dará un aviso cuando se estén utilizando otros puertos.

2.6. Herramientas existentes

2.6.1. PineApple

Cuando hablamos en ciberseguridad de una piña WiFi estamos hablando de una WiFi pineapple, un dispositivo hardware que incorpora una suite de herramientas informáticas con el objetivo de buscar todo tipo de vulnerabilidades en redes inalámbricas.

Este pequeño dispositivo, desarrollado por la empresa Hack5, es el más completo hasta la fecha. Dispone de dos versiones, la versión Nano (figura 2.5) que ronda los 100-150 euros y la versión Tetra (figura 2.6) que ronda los 250-313 euros.

Para quien no pueda pagarse los costos, puede tratar de emular el mismo hardware, decompilar el firmware que te proporcionan desde hack5, y compilarlo para la versión de tu hardware. Este metodo es bastante complejo y lleva su tiempo. La forma de hacerlo es a través del mini router GL.iNet AR-150 y un adaptador WiFi vía USB con el Chipset AR9271 como el que presenta la antena ALFA.

Figura 2.5: WiFi Pineapple Nano

Figura 2.6: WiFi Pineapple Tetra

La creación de esta WiFi Pineapple casera esta poco documentada actualmente. Todos los artículos encontrados en la red otorgan la información para el Firmware 2.0.2. Actualmente la última versión del Firmware es la 2.6.0. Podemos encontrarnos por internet un artículo (Baameiro (2019)) con la información para el montaje casero de una WiFi Pineapple actualizada a la última versión de su Firmware. Este proceso de montado puede llevar desde dos días hasta 1 semana.

Algunos de los modulos más usados y comentados son:

- Deauthentication: mediante ataques de denegación de servicio (DoS) a través de peticiones con el flag DEAUTH activo fuerza que la víctima se desconecte del punto de acceso Wi-Fi al que se ataca.
- NMAP: nos permite realizar escaneos sobre la red a través de herramientas como nmap.
- Tcpdump: a través de la herramienta tcpdump nos permite realizar escaneos por la red.

2.7. Software existente

En la siguiente sección se tiene en cuenta el software que existe en la actualidad para la gestión de la defensa en una infraestructura inalambrica de tipo WiFi.

Todas las herramientas mostradas a continuación presentan un sniffer. Es por ello que las sondas que tenga WiFi-Pocket también implementarán un sniffer.

2.7.1. Horst

- URL: `https://github.com/br101/horst`

- Licencia: GPLv2
- Estado: con soporte

Horst, por sus siglas Highly Optimized Radio Scanning Tool (HORST), es una herramienta que nos permite analizar el tráfico WiFi IEEE802.11. Se trata de un programa libre, con licencia GNU General Public Licence v2.0. Esta desarrollado en C. Podemos encontrar todo su código a través de la url `https://github.com/br101/horst` . Este programa, al igual que otros como nmap, se puede encontrar en los repositorios oficiales de distribuciones conocidas como Debian o Arch. No es tan avanzado como otras herramientas como kismet, wireshark o tcpdump. Entre sus puntos fuertes destaca el poco espacio que ocupa el código en memoria en comparación con otras herramientas por lo que si vamos a usar algún modelo con poca memoria o poca fuerza de procesamiento es una alternativa muy viable. Horst es ideal para modelos con sistema operativo OpenWRT ejecutadas en modelo GL.iNet. También, destaca por parsear de manera más clara y efectiva toda la información. Otro de sus puntos fuertes, es que nos deja realizar la conexión por sockets. Este programa no cuenta con gestión de reglas basadas en los paquetes. Como parte de la investigación, se contactó con el autor para realizar una posible implantación de la herramienta dentro de las sondas de WiFi-Pocket.

2.7.1.1. TCPDump

- URL: `http://www.tcpdump.org`
- Licencia: BSD
- Estado: con soporte

TCPDump es el programa más conocido de análisis de tráfico. Actualmente cuenta con más de 100 colaboradores en su repositorio principal y se encuentra en los repositorios de las principales distribuciones de ciberseguridad.

Es una herramienta más elaborada y pesada que horst, pero más completa. Este programa tampoco cuenta con gestión de reglas basadas en los paquetes.

2.7.1.2. Kismet

- URL: `https://www.kismetwireless.net`
- Licencia: GPLv3
- Estado: con soporte

Kismet, al igual que las dos herramientas anteriores, también es un "sniffer" de paquetes con licencia GPL y escrito en C++. Una de las ventajas que presenta, es que puede funcionar como un Sistema de Detección de Intrusiones vía Wifi, es decir, un WIDS. El funcionamiento es sencillo, necesitando de varias partes. Por un lado, necesitamos una sonda que nos permita capturar paquetes, los cuales serán enviados a un servidor para su interpretación. Por otro lado, un servidor que bien puede ser la propia sonda, para poder interpretar la información.

Debido a su complejidad, dado que no solo está destinado a WiFi sino que incluye más modalidades como BlueTooh, SRD, ..., resulta muy difícil extrapolar

parte de las utilidades que posee. Además, se ha comprobado que no es muy exacto a la hora de detección de ataques. Como el objetivo de WiFi-Pocket es poder dotar de un motor moldeable de reglas, se descarta esta herramienta.

Algunas de esas reglas se pueden encontrar en la siguiente url: `https://www.kismetwireless.net/docs/readme/alerts_and_wids/RL`

2.7.1.3. Snort Wireless

- URL: `http://snort-wireless.org/`
- Licencia: GPLv3
- Estado: sin soporte

Snort es el Sistema de Detección de Instrusos basado en red más conocido. Cuenta con una licencia libre y es totalmente gratuito. Su sencillez emana a la hora de la creación de las reglas que permiten al usuario detectar las diferentes amenazas que hay en una red. Estas reglas son las que definen los patrones que se utilizan a la hora de monitorizar la red.

A su vez, este IDS puede funcionar como sniffer y registro de paquetes. A través de la captura de paquetes, aplicandole una lógica proveniente de las reglas, se generan las alertas.

El formato de las reglas es el siguiente se puede observar en la figura **??**.

Figura 2.7: Snort Rules

Snort posee una comunidad muy grande y se pueden consultar todas las reglas que han creado los diferentes usuarios. Estas reglas se envían a través de la lista de correos y son registradas. Esto permite tener un IDS actualizado.

Surgió una variente a este NIDS denominada WIDS que se centraba en las redes Wireless. Esta alternativa, tal y como se puede consultar en su propia web, no se encuentra operativa.

Capítulo 3

Arquitectura de la Aplicación

"La información es poder. Pero como con todo poder, hay quienes lo quieren mantener para sí mismos."
— Aaron Swartz

WiFi-Pocket es una herramienta que permite realizar tareas ofensivas, de Red Team, y tareas defensivas, de Blue Team. Está compuesta por un panel central, un servidor DNS, un servidor DHCP, varias bases de datos, un servidor web y un conjunto de sondas, que intercambian información a través de su propia red virtual de forma cifrada. A través del panel de control tendremos acceso a varias tareas, entre las cuales se encuentran las acciones de ataque. A su vez, en la parte defensiva, tiene un Sistema de Detección de Intrusos vía WiFi (WIDS) en tiempo real.

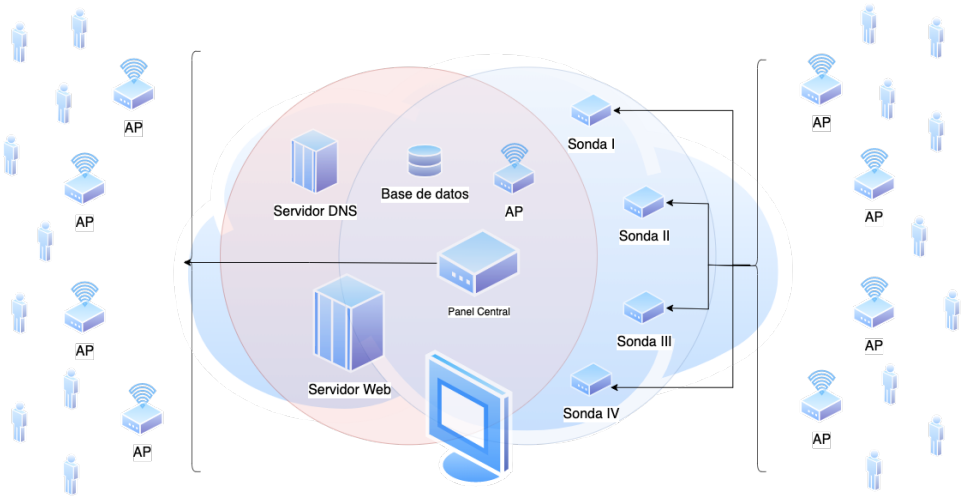


Figura 3.1: Arquitectura de WiFi-Pocket

Como se puede ver en la figura 3.1, la parte ofensiva se muestra como un círculo de color rojo y la parte defensiva se muestra como un círculo de color azul. Rodeando a todo el conjunto de los elementos se representa, con forma de nube, la red privada

virtual. Como puede verse, unos elementos pertenecen a la parte ofensiva, otros a la parte defensiva y algunos a ambos. En el lado derecho de la imagen, aparecen los elementos sobre los que se recoge información a través de las sondas. En la parte izquierda, aparecen todos aquellos elementos sobre los que se pueden llevar a cabo acciones ofensivas. La pantalla que aparece en la parte central inferior representa la conexión por parte de un usuario externo al conjunto que conforma WiFi-Pocket.

3.1. Introducción

El código de la herramienta WiFi-Pocket se aloja en un repositorio de GitHub. Esta herramienta se divide en cinco archivos: control_panel, scripts, sondes, LICENCE y README.

A continuación, se definen los principales:

- control_panel: se trata de la carpeta que contiene todos los recursos de los que hará uso el panel de control. Entre estos recursos, se encuentran imágenes, archivos css, php, html, ...
- scripts: el objetivo de esta carpeta es contener todas las herramientas de las que dispondrá WiFiPocket que le ayudarán en las tareas sobre todo ofensivas. Estas herramientas estarán escritas en Python y Bash.
- sondes: el nombre de esta carpeta hace referencia a las sondas, en inglés probes, de las que hará uso la herramienta para extraer información. Contiene un script que permite realizar la instalación y configuración de las sondas.

Se diferencian dos partes principales en la estructura del repositorio. Por un lado, tenemos el panel de control, que engloba el servidor DNS, el servidor DHCP, las reglas de iptables que se pueden aplicar, el servidor web, el Captive Portal y las bases de datos. Por otro lado, tenemos toda la parte de las sondas. La clave de la comunicación entre ambas partes reside en el formato JSON de envío de los datos.

3.2. DHCP

En la sección 4.10 se define en que consiste un servidor DHCP y cómo funciona. Dentro del código, se cuenta con una guía llamada README.md escrita en Markdown donde se explica paso a paso en qué consiste, cómo editar la configuración y cómo levantar el servidor.

```
# How To
## Install the DCHP Server
```sh
$ sudo apt install isc-dhcp-server
```

Edit the configuration file
You can find this file in the route `/etc/dhcp/dhcpd.conf` . Use
    our template named dhcp.conf contained in this folder.

Start or Stop DHCP services
```

```
```sh
$ sudo systemctl start isc-dhcp-server.service
$ sudo systemctl stop isc-dhcp-server.service
```

Check the DHCP Server
Use a client and put in the terminal:
```sh
$ sudo dhclient
```
```

A su vez, existe a disposición de los usuarios un archivo bajo el nombre de "dhcp.conf" con la configuración por defecto de un servidor DHCP. El contenido del fichero es el siguiente:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.4 192.168.0.255;

    option domain-name-servers 192.168.1.1;
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;

    default-lease-time 86400;
    max-lease-time 172800;
}
```

3.3. DNS

En la sección 4.9 se define qué es un servidor DNS y cómo funciona. Dentro del repositorio de código, se cuenta con una guía llamada README.md escrita en Markdown donde se explica paso a paso cómo editar la configuración y cómo levantar el servidor.

```
# How To
## Installation
```sh
$ sudo apt udpate
$ sudo apt install bind9 bind9utils bind9-doc
```

Use of DNS Server
Please edit the file `/etc/bind/named.conf`, you can use our
    template contained in this folder with name `named.conf`. In
    this template you have the dns server of three social
    networks.
```

Para facilitar a un usuario a que pueda levantar un servidor DNS en el repositorio de código se encuentra un archivo por defecto. A su vez, de muestras archivos de zonas de ejemplo. El archivo se llama "named.conf" y su contenido es el siguiente:

```
    acl rrss {
```

```
        192.168.0.1/24;
      };

      options {
        include "/etc/bind/common/options.common.conf";

        allow-query { trusted; };
        allow-transfer { none; };

        // authoritative only
        recursion no;
      };

      view "rrss" {
        match-clients { rrss; };

        zone "instagram.com" in {
          type master;
          file "/var/named/instagram.com.zone";
        };

        zone "facebook.com" in {
          type master;
          file "/var/named/facebook.com.zone";
        };

        zone "twitter.com" in {
          type master;
          file "/var/named/twitter.com.zone";
        };
      };
```

Se definen varias zonas. Por cada zona, también se deja un fichero de configuración. Un ejemplo de estos ficheros de configuración sería el siguiente:

```
$TTL 300
$ORIGIN twitter.com.

@  1D  IN  SOA twitter.com. admin.garciabaameiro.com (
     1; serial
     3H; refresh interval
     1m; retry
     10m; expiry period
     5m; negative TTL
)
        IN  NS  ns
ns      IN  A   192.168.0.1
www     IN  A   192.168.0.1
```

3.4. Servidor Web

Otro de los aspectos necesarios para la implementación de WiFi-Pocket es un servidor web. Para facilitar la configuración de un servidor web sencillo, se ha optado por Apache. El funcionamiento de este tipo de servidor y la razón de uso del mismo viene definido en la subsección 4.5.2.

Se hace entrega en el repositorio de código de archivos de configuración por defecto de los virtualhost de Apache junto con una guía de cómo realizar esta configuración.

La guía de configuración es la siguiente:

```
# How to
First, create a new virtualhost. You can use our templates that
    are in the folder templates. Plase, name it rrss.conf.

Secondly, we need to enable the virtualhost os rrss, named rrss.
    conf.

```sh
$ a2ensite rrss.conf
$ sudo systemctl restart apache2
```
```

Un ejemplo de virtualhost, sería el siguiente:

```
<VirtualHost *:80>
    DocumentRoot "/templates/www/facebook"
    ServerName facebook.org
    ServerAlias www.facebook.com

    # Other directives here
</VirtualHost>
```

3.5. Iptables

Aunque no consta como uno de los pilares que conforman WiFi-Pocket, sí se hace uso de un firewall. Este firewall es iptables. La idea de emplear un firewall es evitar que puedan descubrir nuestra sonda en una red y puedan atacarla. Dentro del repositorio de código existe un fichero llamado README.md con unas reglas.

```
# Ideas
The firewall rules that you need to redirect all traffic to our
    web server.
```sh
# Rules to stop discovery using ping tests
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
iptables -A INPUT -i eth1 -p icmp --icmp-type echo-request -j
    DROP
iptables -A INPUT -s 192.168.1.0/24 -p icmp --icmp-type echo-
    request -j ACCEPT
```
```

3.6. Panel de Control

Lo encontramos dentro de la carpeta control_panel. Esta carpeta aloja los ficheros de configuración de:

- Servidor web: se encarga de la configuración de los diferentes virtualhost que serán usados por la aplicación.
- Captive Portal: esta carpeta aloja el front-end y el back-end del panel de control, es decir, aloja toda la interfaz de la aplicación.
- Servidor de las sondas: esta carpeta contiene los archivos necesarios para la creación del servidor vía websockets mediante nodejs. También guarda las reglas del IDS utilizadas para analizar la información de los paquetes interceptados por las sondas conectadas a WiFi-Pocket.
- Servidor DHCP: contiene la configuración del servidor DHCP.
- Servidor dns: contiene la configuración del servidor DNS.
- Reglas de Iptables: contiene instrucciones para la aplicación de reglas de protección cortafuegos mediante el programa iptables.
- Templates: esta carpeta pretende alojar los templates que usan para suplantar páginas web en las labores ofensivas de la herramienta.

La configuración de las sondas, el servidor dhcp, el servidor web, las reglas de iptables y los templates se hace de forma manual sin ninguna interfaz web.

3.6.1. Front-End Web

El panel de control a través del Captive Portal presenta un diseño moderno que se adapta a diferentes dispositivos. Esto hace que pueda poder ser usado sin ningún problema por teléfonos móviles, tablets y ordenadores.

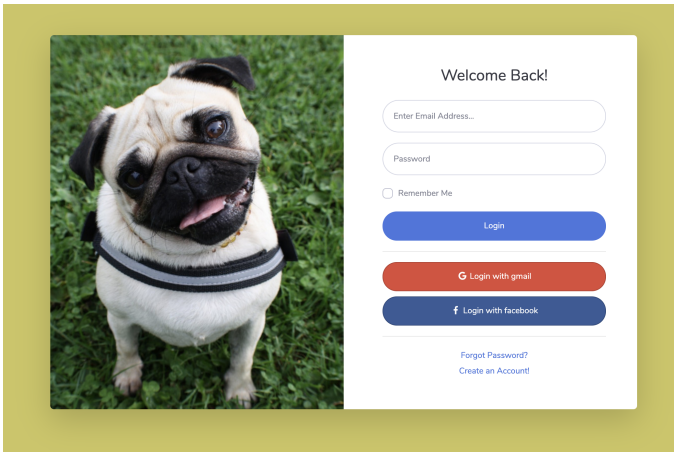


Figura 3.2: Captive Portal de WiFi-Pocket

A través del Captive Portal accedemos al panel de control en el que podemos consultar información y ejecutar tareas.

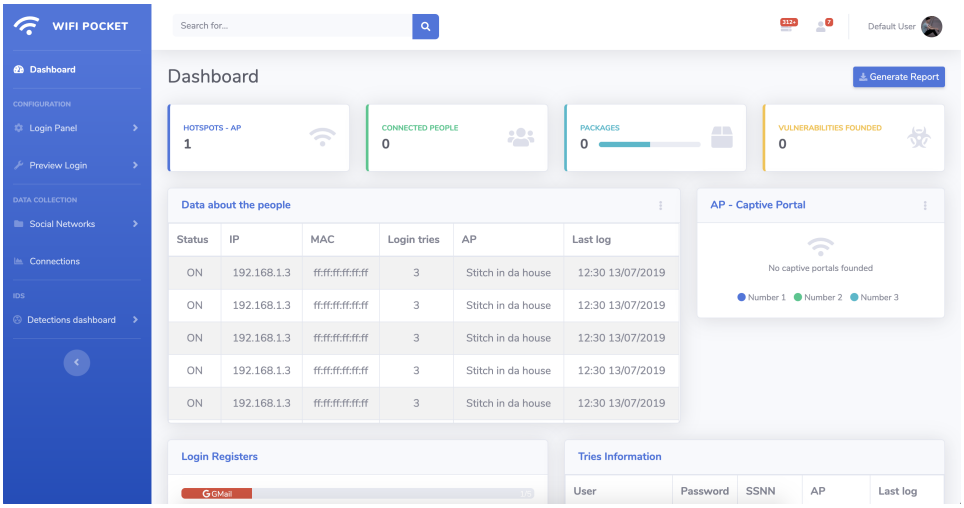


Figura 3.3: Dashboard principal de WiFi-Pocket

Como se puede observar en la figura 3.2, disponemos de un menú situado a la izquierda que nos presenta las diferentes funcionalidades que ofrece WiFi-Pocket. Entre ellas opciones, se encuentra:

- Dashboard: es la interfaz principal que vemos como usuarios. Aquí se muestra información sobre el número de puntos de acceso creados, la gente conectada a los mismos, los paquetes intercambiados, las vulnerabilidades de esos dispositivos, los intentos de login, ... Estas funciones a implementar como trabajo futuro, permiten visualizar en tiempo real que es lo que está pasando.
- Configuration: aquí encontramos la configuración relevante al panel de control. De momento, cuenta con las siguientes las subopciones:
  - Edit Style: nos permite modificar toda la estructura del panel de control basado en su diseño, logo, imágenes, fondo, texto, redes sociales y campos empleados en el formulario.
  - DHCP: pendiente de implementar la interfaz visual, permitiría manejar el servidor dhcp a través de una interfaz web visual.
  - Web Server: pendiente de implementar la interfaz visual, permitiría manejar el servidor web a través de una interfaz web visual.
  - Clone a website: pendiente de implementar la interfaz visual, permitiría clonar sitios web e implementarlos a modo de plantillas a través de una interfaz web visual.
  - Login Panel: nos permite poder visualizar como vería un usuario normal el portal de acceso.
- Data Collection: este apartado permitirá una vez implementado en el futuro mostrar con gráficas los diferentes datos de los usuarios que han sido recolectados. Toda la información se recopilaría de forma masiva.

- Actions: aquí encontramos un panel de acciones que podemos realizar. Dentro de estas acciones se encuentra la creación de puntos de acceso y de listas blancas y/o negras.
  - Create AP: nos permite configurar diferentes puntos de acceso que podremos parar o activar.
  - Manage Users Lists: está subdividida en dos vistas: Whitelist y Blacklist por un lado, y, por otro lado List Users. La primera nos permite crear listas blancas y negras de usuarios y la segunda nos permite controlar y ver todos los usuarios que se encuentran a nuestro alrededor.
- IDS: este apartado nos muestra las opciones de las que dispone el IDS. Este sistema de detección de intrusos es en tiempo real. El subapartado que nos encontramos es:
  - Detection dashboard: este subapartado nos mostrará en tiempo real a pantalla completa los paquetes intercambiados y las alertas que se van generando. Podemos ver los posibles ataques que se están produciendo a nuestro alrededor.
- Attack: este será el panel a través del cual se realizarán labores de Red Team. Nos permite, por un lado, crear diferentes ataques y, por otro, controlar la evolución de los mismos. Este proceso se podrá controlar en tiempo real.

3.6.1.1. Diseño responsive

En la figura 3.3 se muestra el diseño visto desde una pantalla de ordenador. La idea de WiFi-Pocket es ser ejecutado en cualquier dispositivo, ya sea móvil o tablet.

La visualización del panel de control a través de un iPad puede verse en la figura 3.4. También podemos ver como se mostraría en el caso de que el usuario emplease un iPhone en la figura 3.5.

3.6.1.2. Pop Ups

Antes de realizar cualquier cambio o ejecutar cualquier tarea se pregunta al usuario si está seguro de lo que está haciendo. En caso de estar guardando una configuración, como se puede ver en la figura 3.6, se le pregunta si está seguro de querer guardas los nuevos cambios.

3.6.1.3. Notificaciones

A través de WiFi-Pocket realizamos labores ofensivas y defensivas a través de las cuales, en caso de no tener control sobre las mismas, podríamos causar posibles daños en la infraestructura a proteger. Por ejemplo, si no se informa al usuario de que no se han producido cambios, puede estar cambiando en tiempo real el diseño del Captive Portal. Estos cambios pueden indicar peligro al usuario que los ve, provocando que fallen nuestras labores como Red Team.

Para que el usuario que utiliza WiFi-Pocket esté informado de si lo que va realizando se ejecuta con normalidad, no se ejecuta o ha habido problemas, en la esquina inferior derecha aparecen notificaciones informando de ello.

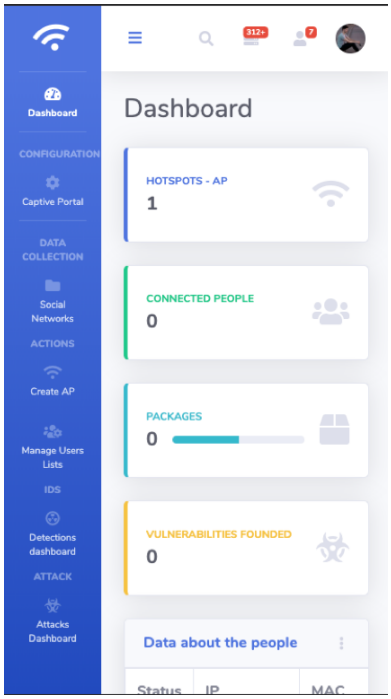


Figura 3.4: Panel de control visto desde un iPad

Un ejemplo de notificación que se emplea cuando el usuario cancela una acción viene reflejado en la figura 3.7.

Otro ejemplo de notificación, en este caso de acción completada, viene en la figura 3.8.

3.6.1.4. Dinamismo

Para evitar que un usuario tenga que salirse de la aplicación o página con el objetivo de refrescar los datos, se ha implementado código Javascript que ya no solo muestre lo que se va realizando sino que también realice las conexiones entre el cliente y el servidor. Estas conexiones se hacen sin refrescar la página.

Un ejemplo de ello, son los campos del formulario del Captive Portal. El usuario que utilice el panel de control puede, sin salirse de la página, agregar nuevos campos o eliminar los que ya hubiera o este creando. Para guardarlos, se utiliza el botón de "save changes".

Para realizar la comunicación entre el cliente y el servidor sin salirse de la aplicación, se emplea AJAX. Para facilitar las cosas, se ha usado AJAX a través de jQuery. A continuación, se muestra una petición AJAX implementada a través del framework jQuery:

```
$.ajax({
    type: "POST",
    url: "actions/ajaxpetitions.php",
    dataType: 'json',
    data: {
```

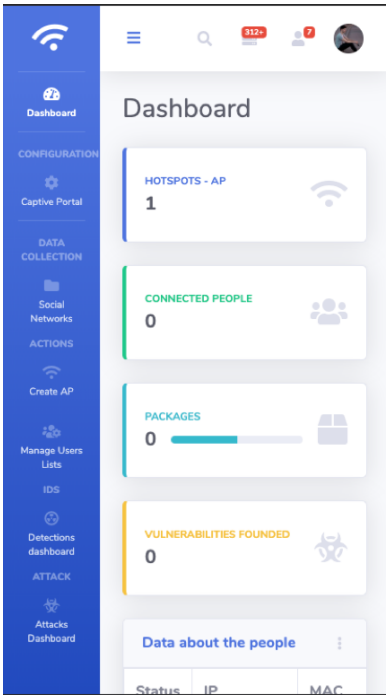


Figura 3.5: Panel de control visto desde un iPhone

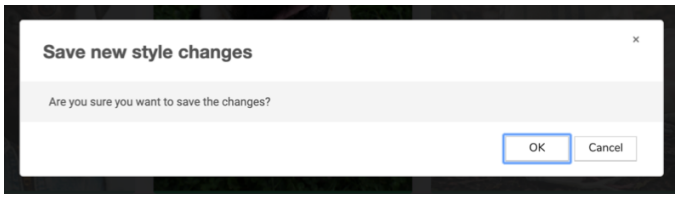


Figura 3.6: PopUp confirmación de accion

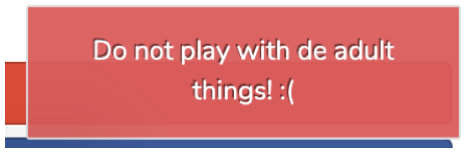


Figura 3.7: Notificación de cancelación de una acción

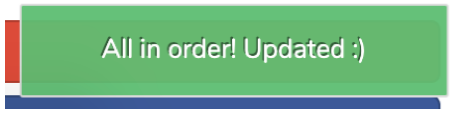


Figura 3.8: Notificación de acción realizada con éxito

```
        type: "update",
        conf: "image",
        id: id
    },
    success: function(){
        alertify.success('All in order! Updated :)');
        document.getElementById('choose_conf').setAttribute('
            value', id);
    },
    error: function(){
        alertify.error('Something was wrong :(');
    }
});
```

Todas las peticiones que realiza WiFi-Pocket a través de AJAX son por POST e intercambian la información en formato json. En caso de que haya habido algún error en la petición, lanzamos una notificación de que algo ha ido mal. Si todo ha ido bien, se avisa al usuario y se modifican los datos con la nueva información.

El fichero que recoge todas las peticiones e invoca a sus respectivas funciones en php se llama ajaxpetitions.php y se encuentra en la carpeta de actions. En el caso de la petición AJAX anterior recogemos los datos como se indica a continuación:

```
if($_POST["type"] == "update")
{
    if($_POST["conf"] == "image"){
        echo json_encode(updateImage($_POST["id"]));
    }
}
```

Este fichero, a su vez, llama a la función que realiza el trabajo solicitado. Esta función se encuentra en el fichero actions.php de la misma carpeta. En este archivo, las funciones se encargarán de buscar el objeto que se desea modificar, hacer petición a la base de datos, y devolver la información.

3.6.2. Back-End

Para la parte del back-end de la aplicación se han implementado todos los archivos haciendo uso de php. Las bases de datos se gestionan a través de la carpeta bbdd-model. Como mencionabamos en el apartado anterior , las acciones requeridas a través de llamadas mediante AJAX son recogidas en la carpeta action, concretamente por el archivo ajaxpetitions.php. Este archivo lanzará órdenes sobre el archivo actions.php.

La carga de ficheros está dividida en submódulos, de tal forma que diferentes vistas pueden cargar un mismo módulo sin suponer una carga sobre el código de la aplicación. Más adelante se explica en el apartado esta estructuración del código.

3.6.2.1. Sesiones de login

El acceso a WiFi-Pocket se hace a través de un usuario y una contraseña. La forma de mantener toda la parte privada oculta del resto de usuarios es a través de sesiones. En cada archivo PHP se comprueba que la sesión sea correcta. En caso de no serlo, se redirige al usuario al login para que realice el inicio de sesión correcto.

A continuación, se muestra una parte del fichero sessions.php donde se manejan las sesiones

```
    $email = $_POST['email'];
    $password = $_POST['password'];

    $user = UserDAO::login($email, $password);

    if ($user)
    {
      // Si esta en la base de datos
      $_SESSION['loggedin'] = true;
      $_SESSION['email'] = $user->getEmail();

      header('Location: index.php');
    }
```

3.6.2.2. Estructura modular

Con el objetivo de facilitar la lectura de código y así incrementar la participación de la comunidad del software libre, se ha estructurado el código a través de diferentes módulos. Todas las vistas siguen un mismo patrón, y cargan los módulos según su tipo.

A continuación, se muestra el código de una de las vistas del panel de ataques. En la parte superior, se ve el manejo de sesiones y la redirección hacia el panel de login en caso de sesión incorrecta. En la mitad del código, en php, se cargan los diferentes módulos referentes al panel. Como resultado tenemos un fichero de pocas líneas de fácil lectura. Si alguna persona desease modificar secciones concretas, solo tendría que ir al archivo especificado.

Esta estructura modular se puede ver en la figura .

```
<?php
  session_start();

  if((isset($_SESSION['loggedin']) == false) || ($_SESSION['
      loggedin'] == false)){
    header('Location: login.php');
  }
?>

<!DOCTYPE html>
<html lang="es">
  <?php include 'modules/head.htm'; ?>
    <body class="bg-gradient-primary">
        <!-- Page Wrapper -->
        <div id="wrapper">

            <?php
                include 'modules/panel/sidebar.htm';
                include 'modules/panel/topbar.htm';
                include 'modules/panel/panel_attacks_logs.htm';
                include 'modules/panel/footer.htm';
```

```
            ?>

        </div>

    <script type="text/javascript" src="assets/js/jquery.min.js"
        ></script>
    <script type="text/javascript" src="assets/js/bootstrap.min.
        js" ></script>
    <script type="text/javascript" src="assets/js/main.js"></
        script>
    <script type="text/javascript" src="assets/js/color-picker.
        js"></script>
    <script type="text/javascript" src="assets/js/ajax_log_conf.
        js"></script>
    <script type="text/javascript" src="assets/js/alertify.js"
        ></script>
    </body>
</html>
```

3.6.3. Funcionalidades

A continuación, se muestran las diferentes funcionalidades con las que cuenta WiFi-Pocket. Estas funcionalidades son accesibles a través del panel de control.

3.6.3.1. Dashboard

Una vez entramos en la aplicación vía conexión web, accederemos a la interfaz principal de WiFi-Pocket. Esta interfaz aparece bajo el nombre de "Dashboard".

Como podemos ver en la figura 3.10, la parte superior cuenta con un botón azul denominado "Generate report", cuyo propósito es exportarnos un fichero pdf a modo de informe con toda la información que se ha ido recogiendo.

Por debajo de este, se encuentran 4 paneles que presentan de forma rápida la información más interesante extraída de la recolección de los datos. Esta información que se está recogiendo pertenece a los puntos de acceso creados, a las personas que hay alrededor, a los paquetes que se han interceptado y a las posibles vulnerabilidades que se hayan encontrado.

Debajo de estos cuatro paneles nos encontramos a la izquierda un apartado, un apartado donde se muestra en tiempo real los datos registrados y, a la derecha, un mini panel con información de los puntos de acceso levantados.

Dentro de la misma vista, hacia el final de la misma, nos encontramos 3 paneles, separados en dos columnas. Por un lado, viendo la figura 3.11, en la parte izquierda se encuentran los intentos de login

que han tratado de hacer los usuarios en las diferentes redes sociales que hemos falseado a través de nuestro Captive Portal. Este apartado queda pendiente como "Trabajo futuro" y pretende mostrar los intentos de login que nos permiten obtener la contraseña que nos dará acceso a la red social (verdadera) del usuario que ha intentado iniciar sesión.

Por otro lado, en la parte derecha, se muestra los usuarios que tenemos alrededor, que han sido conocidos tras realizar escaneos de red, junto con un historial

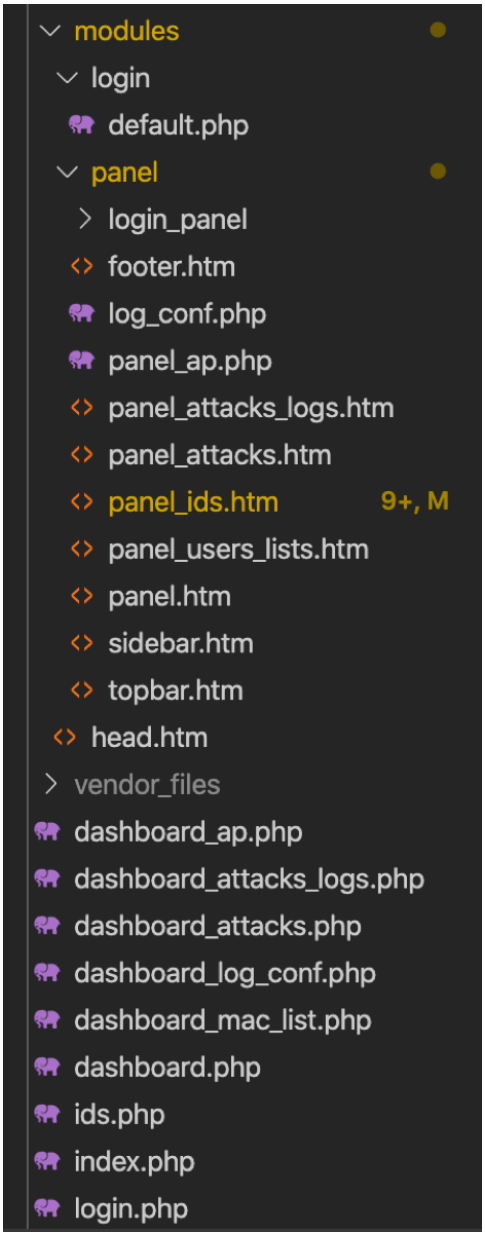


Figura 3.9: Parte II del Dashboard de WiFi-Pocket

de registro de los intentos de acceso a nuestro Captive Portal principal donde se encuentra el panel de control.

3.6.3.2. Captive Portal

En la parte derecha de la aplicación, nos encontramos un menú fijo que nos acompañará en todo el recorrido de uso de WiFi-Pocket y que nos permitirá con-

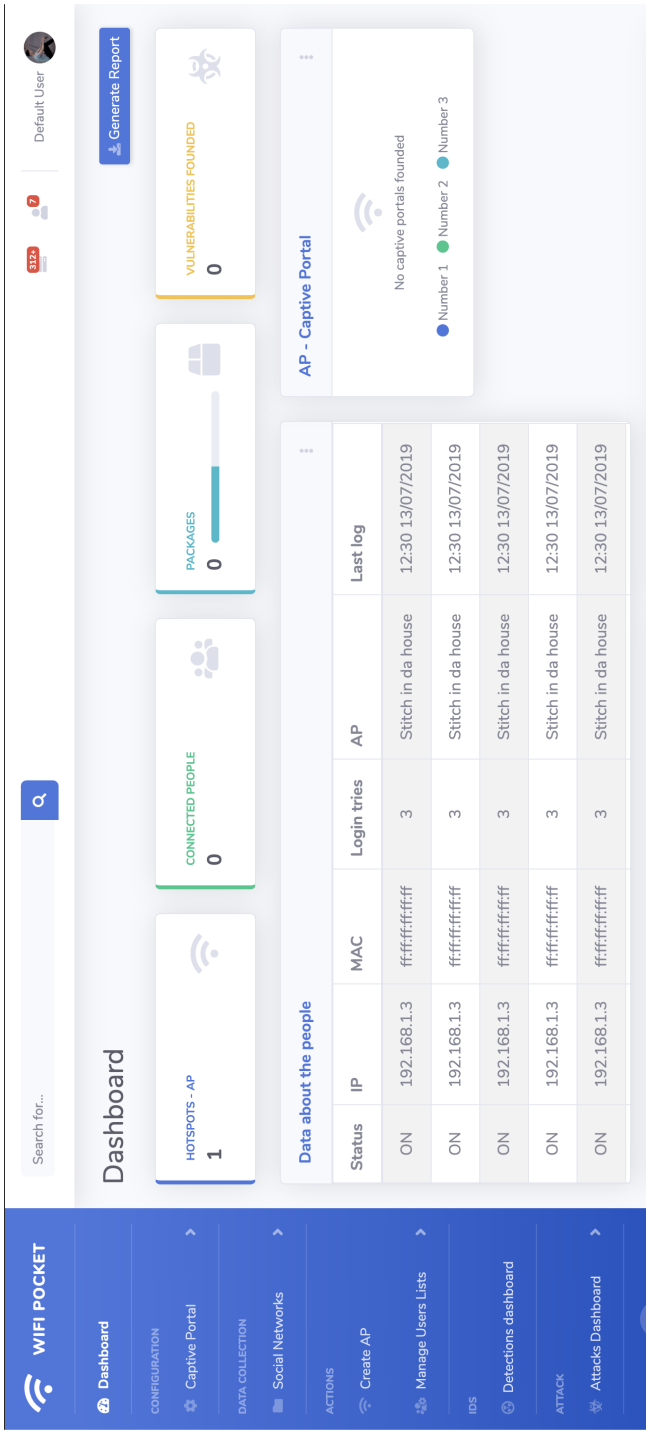


Figura 3.10: Parte I del Dashboard de WiFi-Pocket

Figura 3.11: Parte II del Dashboard de WiFi-Pocket

figurar el Captive Portal. En este apartado vamos a explicar cómo modificar esta configuración.

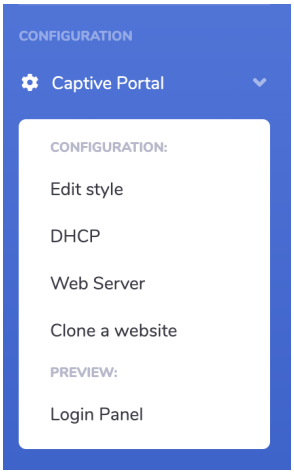


Figura 3.12: Menú donde aparece la configuración del Captive Portal

Desplegando esta opción, nos encontramos con diferentes opciones. De momento, pendientes de implementar en un "Trabajo Futuro" se encuentran la gestión de DHCP y del Servidor Web vía interfaz web y no vía comandos. También, se ha dejado como trabajo futuro el poder realizar clonaciones de las páginas web que el usuario requiera para posteriormente ser modificadas y adaptadas como Captive Portal o designadas a redes sociales de las cuales extraer información.

Por lo tanto, las opciones que están implementadas y nos interesan son "Edit style" y "Login Panel". "Edit style" nos permite modificar el Captive Portal y la interfaz que verán los usuarios antes de acceder al panel de control. "Login Panel" nos permite ver los cambios que hayamos implementado, como por ejemplo, modificaciones en el diseño del propio Captive Portal, en el logotipo que aparecerá en la portada, en los campos de datos que el usuario debe introducir y en los campos de las redes sociales a través de las cuales un usuario puede iniciar sesión. Estos cambios anteriormente mencionados se han denominado "subapartados".

El apartado "Edit style" puede verse en la figura 3.13.

Todos estos cambios se guardan por subapartados si el usuario así lo desea (hay un botón habilitado para ello).

En una primera instancia, nos encontramos que podremos modificar la estructura que se muestra al acceder al login. Esta estructura tiene un template por defecto y un template denominiado "Start Up". A modo de "Trabajo Futuro" ha quedado pendiente la posibilidad de poder subir los templates que el usuario desee. Estos deben tener la estructura necesaria para poder realizar los cambios.

El subapartado Logo nos permite editar el logo principal que aparece en el Captive Portal. En el template por defecto no se usa ningún logo, por tanto, esta opción se encuentra deshabilitada.

Siguiendo con el resto de las opciones, como podemos ver en la figura 3.14, nos encontramos el subapartado estilo (style). Este subapartado nos permite modificar el fondo que aparecerá en el Captive Portal. La modificación del fondo, se ha implementado de tal forma que el usuario puede ver con antelación el color que ha

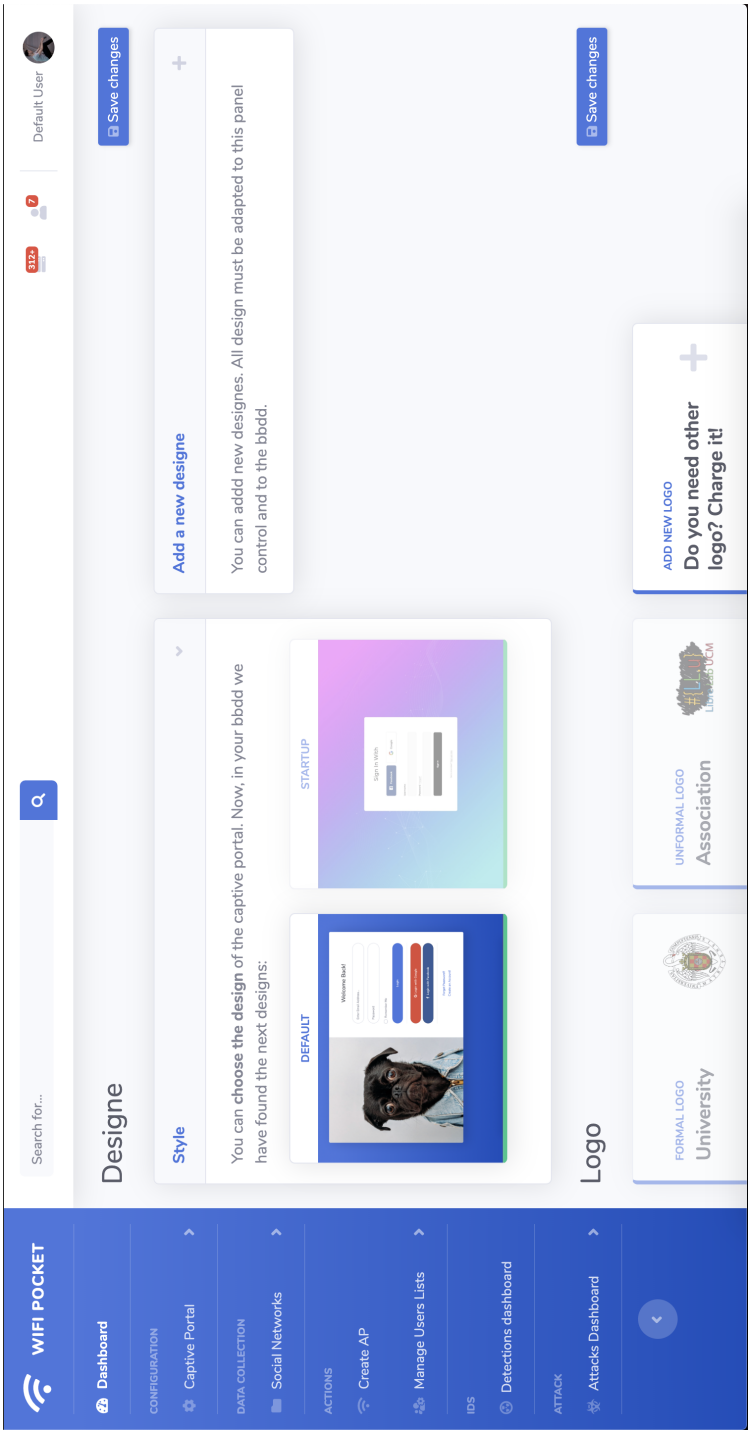


Figura 3.13: Configuración del diseño del Captive Portal

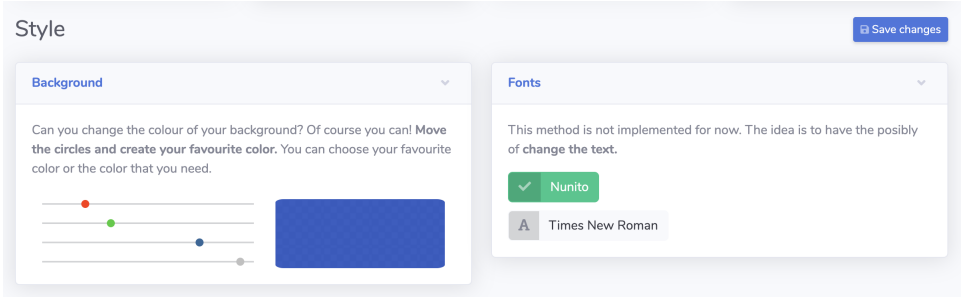


Figura 3.14: Configuración de las fuentes de texto y del color del fondo del template por defecto

elegido y puede regularlo a su gusto.

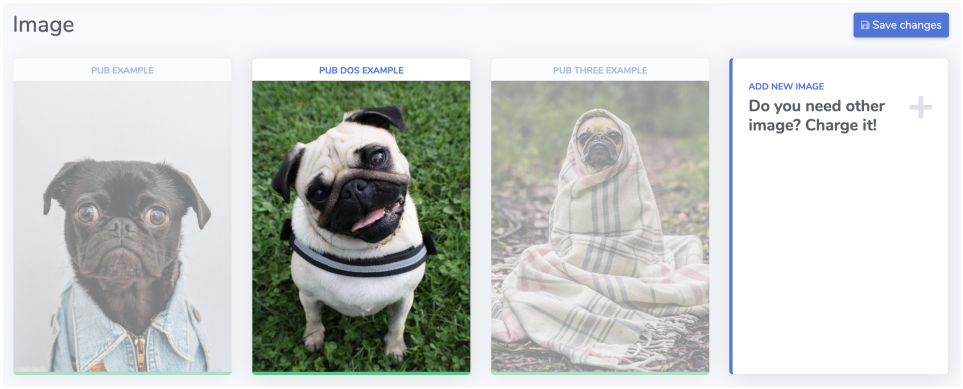


Figura 3.15: Configuración de las imágenes del template por defecto

También se puede cambiar la imagen principal que presenta el template (ver figura 3.15). Para el presente template, a modo de ejemplo y, por defecto, se han utilizado imágenes de perros de raza Pug.

Finalmente, en la parte inferior de la ventana de edición del Captive Portal, podemos recabar datos vinculados al login de un usuario. A la derecha de este subapartado, podremos modificar los campos que ya hay de petición de datos al usuario, tanto eliminando los que consideremos que no son necesarios como añadiendo nuevos. Se ha implementado de forma que cualquier modificación se va mostrando en directo en la pantalla. Estos campos pueden ser de tipo “text” o de tipo “password”.

A su derecha, contamos con númerosos campos de login de redes sociales. Estos campos pueden marcarse o desmarcarse. Notamos que se han marcado o desmarcado por su color. El color es el propio que representa a la red social en cuestión. En total hay 16 redes sociales. Concretamente están: Google, Facebook, GitHub, Twitter, Instragram, Dropbox, LinkedIn, Flickr, Microsoft, Reddit, Pinterest, Soundcloud, Tumblr, Vimeo, VK y Yahoo.

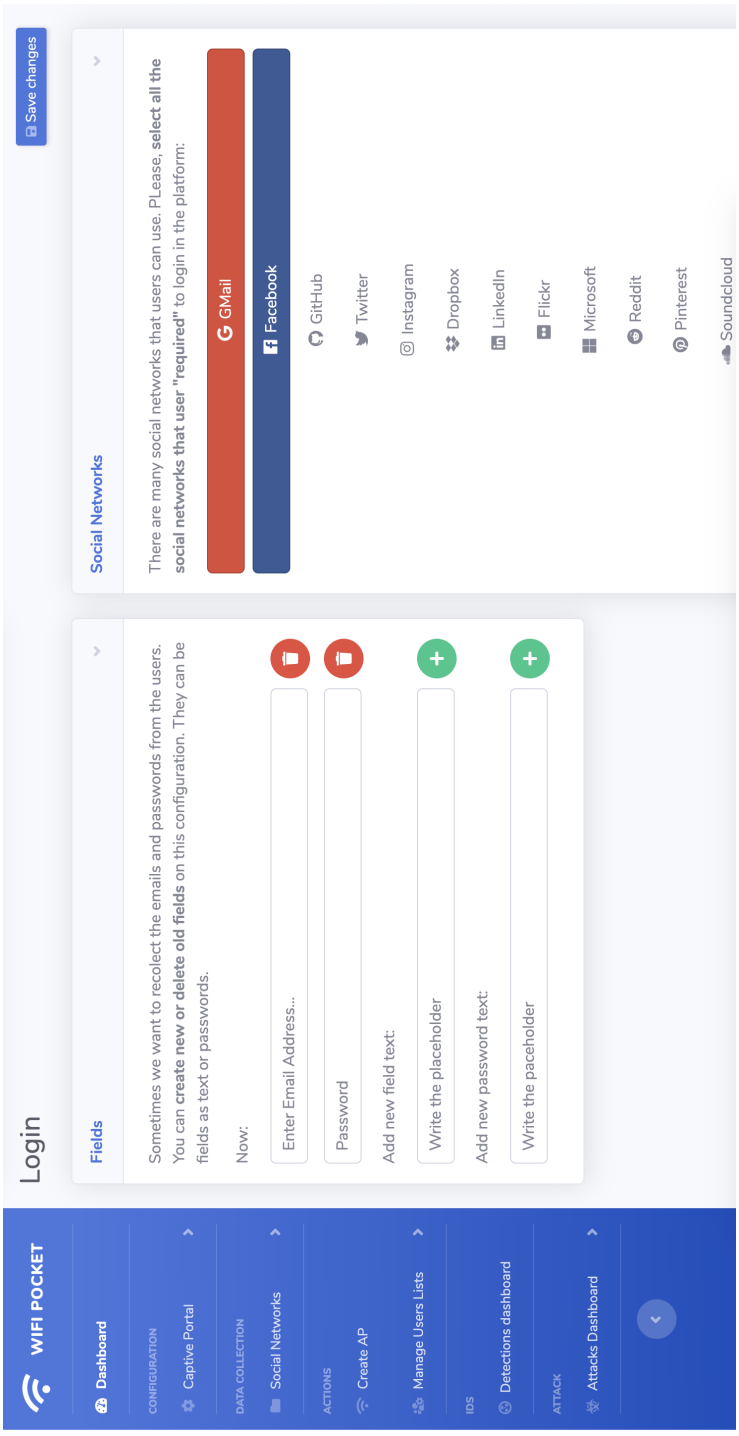


Figura 3.16: Configuración de los campos de login y de las redes sociales para iniciar sesión del template por defecto

3.6.3.3. Data collection

La segunda opción que nos encontramos en el menú principal de la interfaz es "Data Collection". Este apartado fue creado para recopilar toda la información personal que vayamos recogiendo de los usuarios.

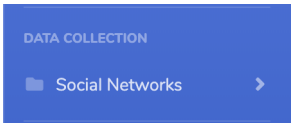


Figura 3.17: Menú donde aparece la recolección de información de WiFi-Pocket

Este apartado ha sido explicado en el punto 3.6.1 y está pendiente de ser desarrollado en un futuro.

3.6.3.4. Actions

Otro de los puntos del menú y, por ende, de la aplicación, es el apartado "Actions". Este apartado contiene una serie de acciones que no están directamente vinculadas con el apartado de mecanismos de defensa ni de mecanismos de ataque.

Estas acciones son "Crear un punto de acceso" y "Manejar listas de usuarios" (figura 3.18)

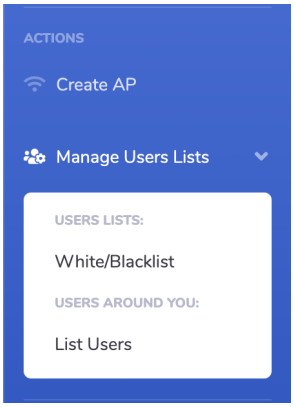


Figura 3.18: Menú donde aparecen las acciones a realizar

Dentro del apartado "Access Point" podremos visualizar (figura 3.19) todos los puntos de acceso que se han ido creando. Desde este panel, podemos pararlos o volver a levantarlos. A su vez, si queremos una nueva configuración y tipo de access point, podemos crearlo.

La configuración que nos permite especificar el nombre del punto de acceso, sobre qué interfaz se volcará, si se tratará de un punto de acceso abierto o cifrado, en caso de ser cifrado qué contraseña tendrá y, por último, que canal se va a emplear.

La opción de cifrado por WPA3 está deshabilitada debido a que es te cifrado es poco utilizado en la actualidad. Una vez guardados los cambios nos aparecerá

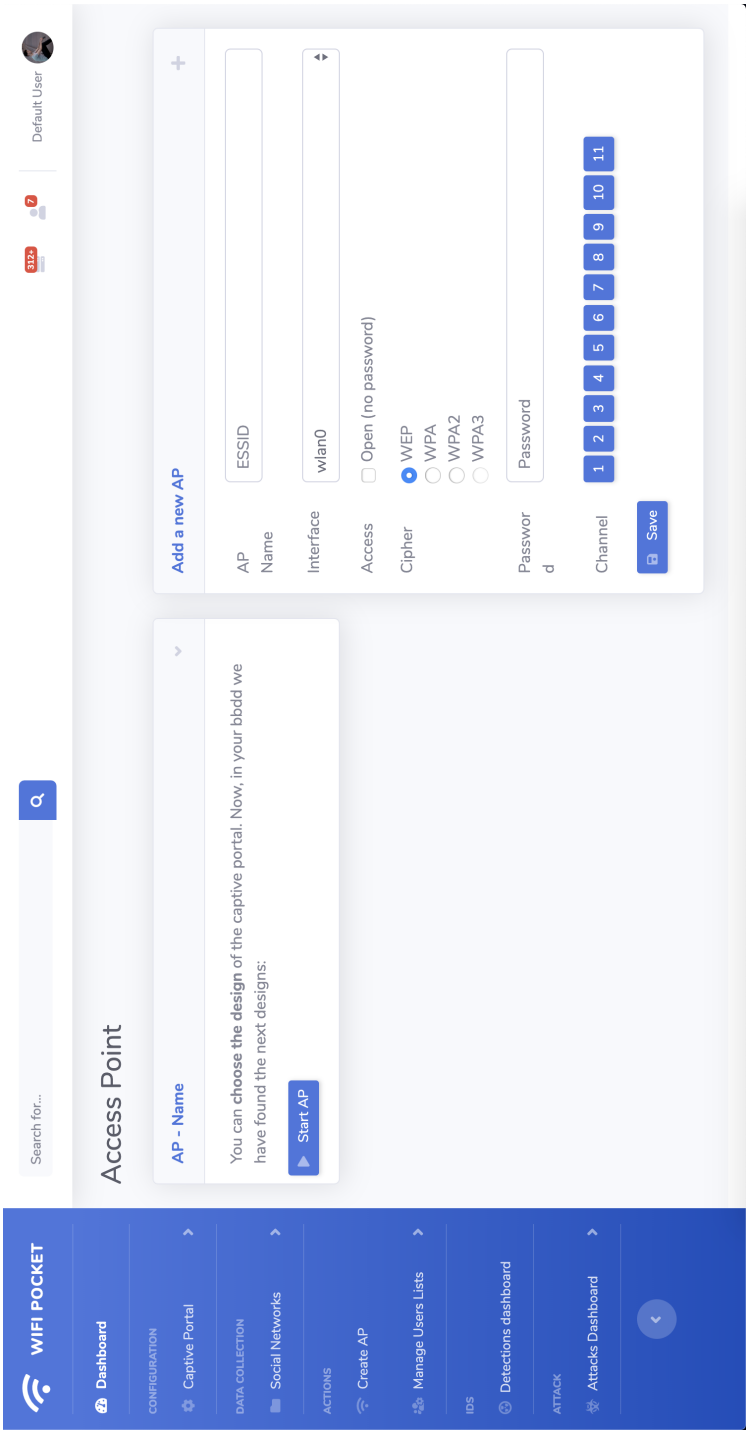


Figura 3.19: Apartado de creación de puntos de acceso

una nueva "caja" con la información de esta configuración. A través de esta nueva "caja" podremos levantar el punto de acceso o, si está activo, pararlo.

Otras de las opciones es manejar la lista de los usuarios.

A continuación, se muestra la gestión de las listas blancas y negras de los usuarios conectados a nuestra red.

Estas listas nos avisan de si un usuario conectado está en nuestra lista negra o si se han conectado usuarios que no están en nuestras listas blancas. Sin embargo, estas listas no impiden que un usuario se pueda conectar y no dejan que podamos desconectarle desde WiFi-Pocket. Estas acciones se han dejado como trabajo futuro.

Una de las acciones que se podría realizar es el seguimiento de un objetivo en base a los paquetes que interceptan las sondas. A través de lo capturado por las sondas conoceríamos los movimientos que realiza el objetivo o en qué lugar de la institución se encuentra.

Podemos ver todos los usuarios que están en la sonda recibe los paquetes o a qué red están conectados los usuarios en el apartado "List Users".

3.6.3.5. IDS Dashboard

Uno de los núcleos fuertes de WiFi-Pocket es su orientación a Blue Team. Esta utilidad, poco investigada en el campo de la ciberseguridad y el mundo del software libre, se presenta en nuestra herramienta de forma visual a modo de panel con la información que puede ser más relevante para una persona que trabaje en ciberdefensa (Blue Team).

Esta información se muestra en varios subapartados (ver figura 3.22). En la parte superior hay dos subapartados que indican el número de usuarios y el número de puntos de acceso que hay (alrededor de las sondas) cuando estamos realizando labores de defensa en seguridad, necesitamos conocer a cuántas personas podría afectarles un ataque o la complejidad de la infraestructura. Por ejemplo, podríamos saber que un ataque se podría realizar sobre 343 usuarios, y que tenemos que ser conscientes de la defensa de 7 puntos de acceso. Si vemos que hay más puntos de acceso que escapan a nuestro control, algo malo puede estar pasando. Son mini paneles con información vital.

Debajo de cada uno de estos subapartados, con un ancho igual pero una altura superior, se muestra información acerca de los paquetes que se están analizando en tiempo real así como información acerca de los puntos de acceso próximos. La información de los paquetes que se reciben nos permite ver la cantidad de información que se está intercambiando. La información de los puntos de acceso nos permitiría, por ejemplo, saber si hay dos puntos de acceso iguales, o si hay dos puntos de acceso con el mismo nombre pero distinta dirección MAC. Comunmente se conocen las direcciones MAC de los puntos de acceso de la infraestructura. Por lo tanto, se podría saber, por ejemplo, cuál es el punto de acceso que ha levantado un atacante a través de este panel.

Finalmente, en el medio y con un ancho mayor que el resto de columnas, se encuentra el panel principal del IDS, dividido en dos subapartados. El que se encuentra en la parte superior, nos muestra de forma visual el estado de alerta que se está produciendo. Este estado de la alerta, dispone de 3 niveles:

- Nivel 1: de color verde, nos indica que va todo bien.

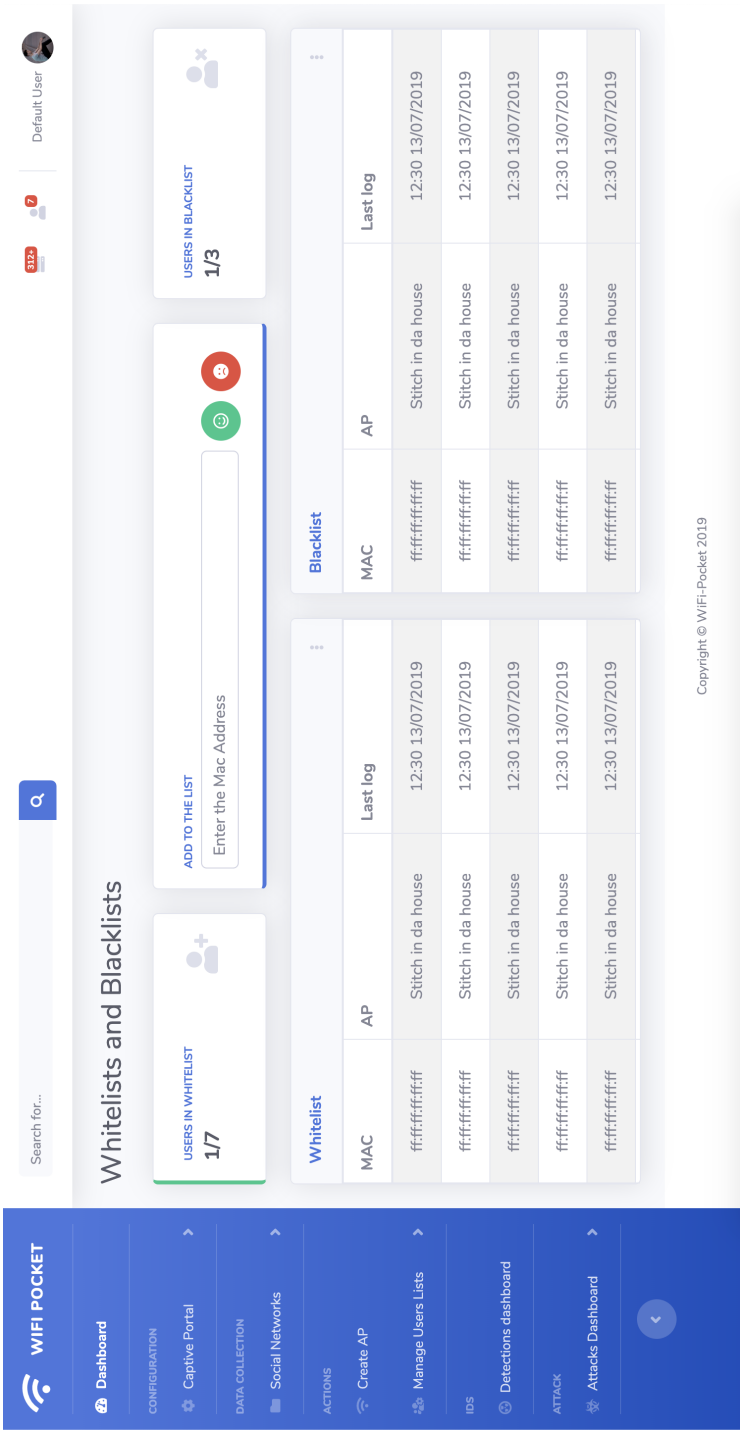


Figura 3.20: Apartado con las listas blancas y negras de WiFi-pocket

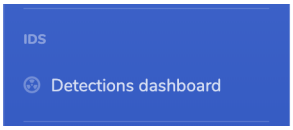


Figura 3.21: Menú donde aparece el dashboard del IDS de WiFi-Pocket

- Nivel 2: de color amarillo, nos indica que hay usuarios que podrían recibir ataques poniendo en riesgo nuestra infraestructura.
- Nivel 3: de color rojo, nos indica que se está produciendo un ataque.

Como decimos, el análisis se hace en tiempo real, por lo tanto, se podrían tomar acciones a tiempo real evitando cualquier daño sobre nuestra infraestructura.

En la sección se explica el funcionamiento en código del IDS con su respectivo impacto en el panel web.

3.6.3.6. Attack Dashboard

Otro de los pilares de WiFi-Pocket es su panel para la realización de ataques. Este panel es accesible a través del menú situado a la izquierda de la interfaz de WiFi-Pocket (figura 3.23).

Para esta vista, disponemos de tan solo 4 subapartados divididos en dos columnas. A nuestra izquierda, el primer subapartado nos permitirá seleccionar el ataque que queremos realizar. Una vez elegido, en el subapartado de la parte inferior denominado "Configuration attack", nos aparecerán los campos necesarios para poder realizar el ataque. Estos campos varian dependiendo del ataque que queramos realizar. Si no hemos rellenado los campos obligatorios, no podemos darle al botón de "play" situado en el primer subapartado.

En la parte derecha de la subdivisión en dos columnas podemos seleccionar la red sobre la cual realizaremos el ataque y, debajo de esta, un panel que nos indica el estado del ataque al momento de darle a "play". Pero, si no indicará que algo ha fallado debido, por ejemplo, a dependencias mal instaladas, falta de paquetes, errores al realizar el ataque, ... Si nos aparece en verde, indicará que se ha podido lanzar el ataque sin ningún problema.

Los ataques que nos permite realizar WiFi-Pocket van desde desautenticar usuarios, escanear las redes de nuestro alrededor previa conexión con ellas, realizar ataques a redes enterprise o realizar ataques para la obtención de claves de los puntos de acceso de alrededor.

Estos ataques llevan una lógica de aplicación por detrás bastante compleja, pero el usuario tan solo tendrá que rellenar campos y lanzar el ataque. Eso es uno de los puntos fuertes de WiFi-Pocket. Podemos, desde el móvil, conectarnos al panel, realizar ataques, observar su proceso, obtener los resultados.

Al realizar un ataque, se guardará un registro del mismo. Todos los registros pueden visualizarse a través de la vista "Log Attack".

Esta vista tan solo dispone de dos columnas. Su objetivo principal es tener un control de todos los ataques que se van produciendo o ya se han realizado.

La primera columna, situada a la izquierda, resalta por sus 4 subapartados de diferentes colores. Estos subapartados muestran el total de los ataques que han sido

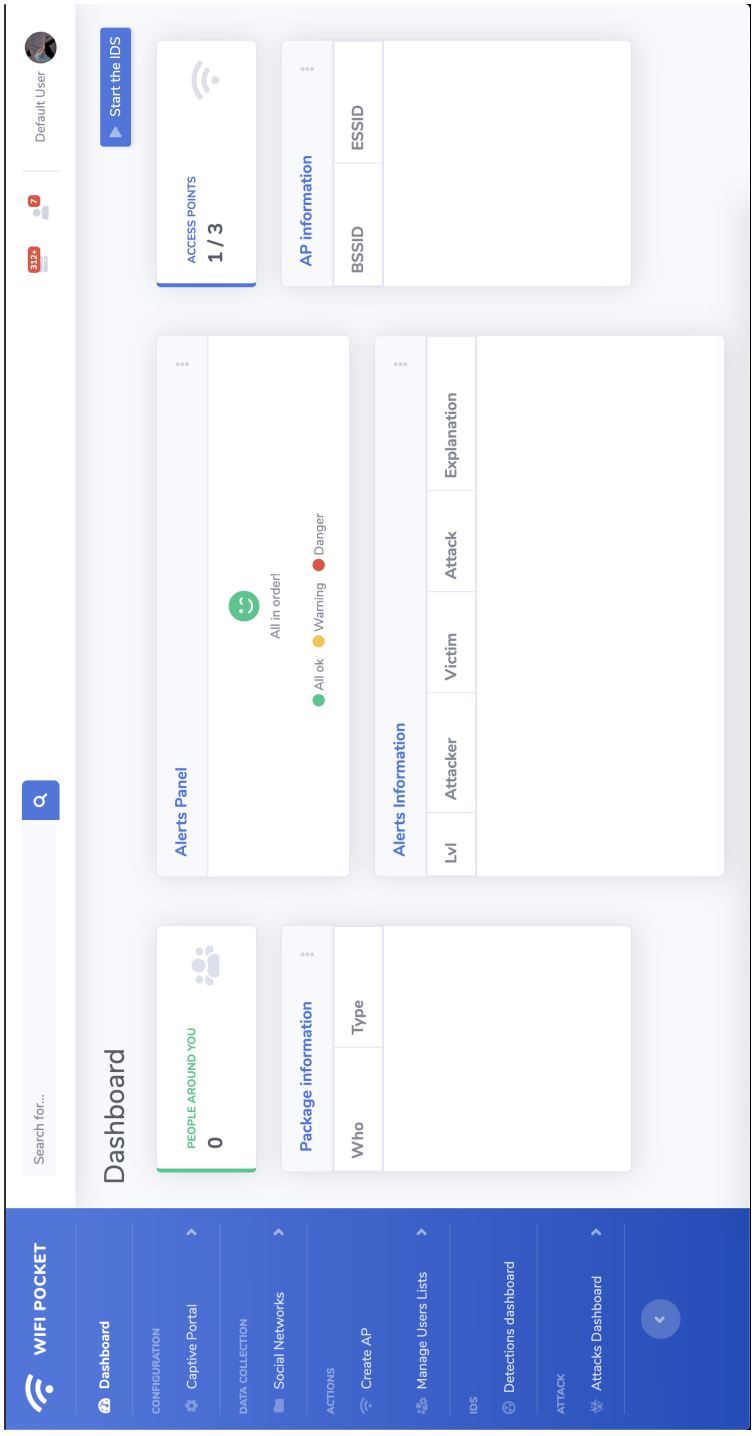


Figura 3.22: Dashboard principal del IDS de WiFi-Pocket

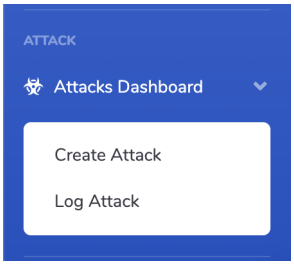


Figura 3.23: Menú donde aparece el apartado de ataque de WiFi-Pocket

lanzados, los que han generado advertencias, aquellos que han dado error y aquellos que se han completado satisfactoriamente.

La segunda columna, situada a la derecha, muestra un registro de los ataques que han sido lanzados con información útil que servirá para detectar la razón de error que se haya producido o la advertencia.

En la sección se explican más en profundidaz algunas de las caracteristicas de los ataques.

3.7. Recolección de información por sondas

3.7.1. ¿Qué es una sonda?

Consideramos sondas a los servidores implementados bajo hardware específico diseñados con el objetivo de realizar las tareas de recolección de información de diferentes redes de su alrededor.

Un ejemplo de sonda que hemos empleado para el presente proyecto se puede ver en la figura 3.27. Esta sonda ha sido creada con una Raspberry Pi 2, dos antenas WiFi, un led azul y una carcasa impresa en 3D presonalizada.

3.7.2. Estructura

Para poder tener activas las sondas vamos a depender de dos tarjetas de red como mínimo. Este requisito se comprueba a través del script de instalación. La idea es poder recolectar datos con una de ellas y con la otra poder conectarnos a una red para enviarlos.

El software de WiFi-Pocket es ejecutado a través de un demonio, creado e integrado mediante el script de instalación, que se lanza al arranque del sistema operativo. Esto nos permite que si hay un corte de red, con tan solo encender el dispositivo y sin pantalla ni teclados, se pueda reanudar la actividad sin ningún problema.

3.7.3. Conexión

La conexión con las sondas varía dependiendo de si lo que queremos es recoger datos (que serán emitidos) o recibir órdenes. Para la emisión de datos se utiliza u servidor de sockets puesto por defecto en el puerto 4000 y ejecutado a través de

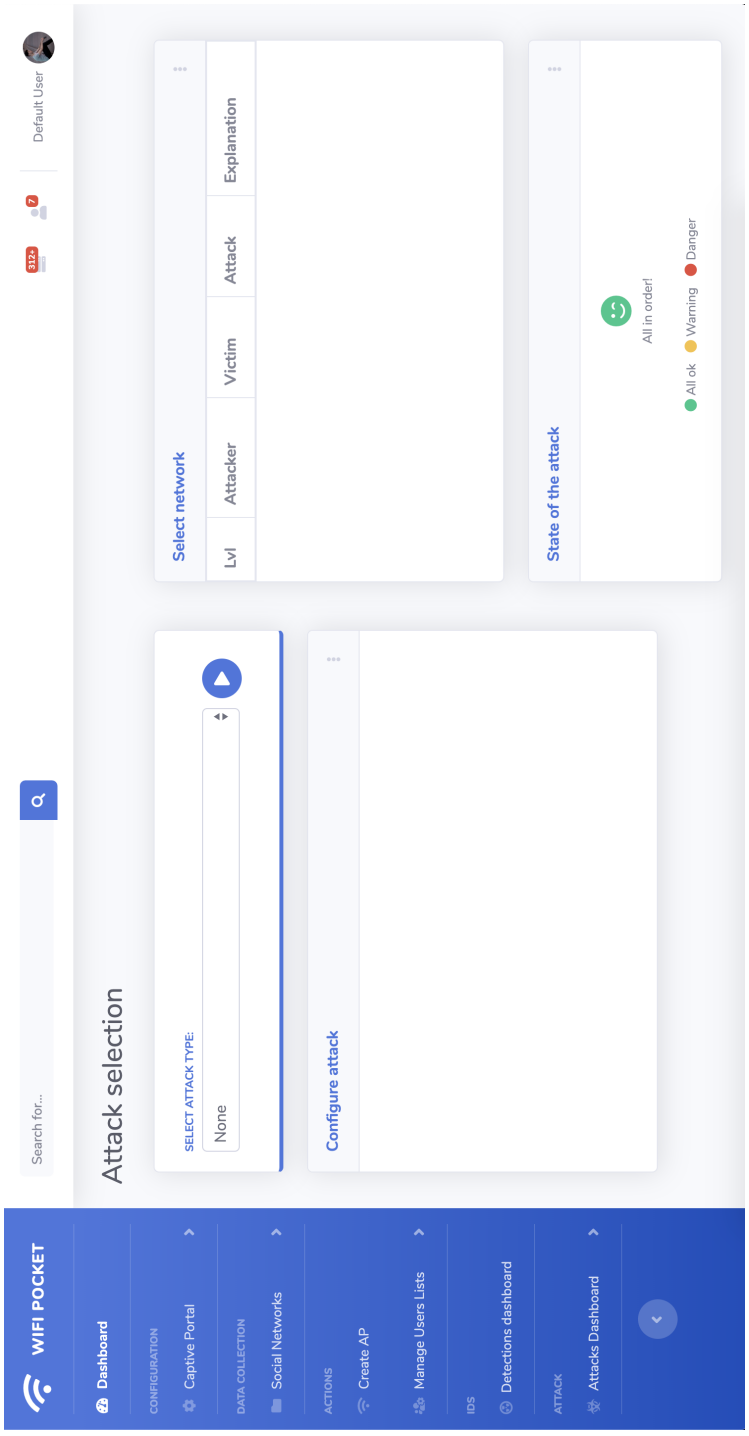


Figura 3.24: Apartado para la realización de ataques de WiFi-Pocket

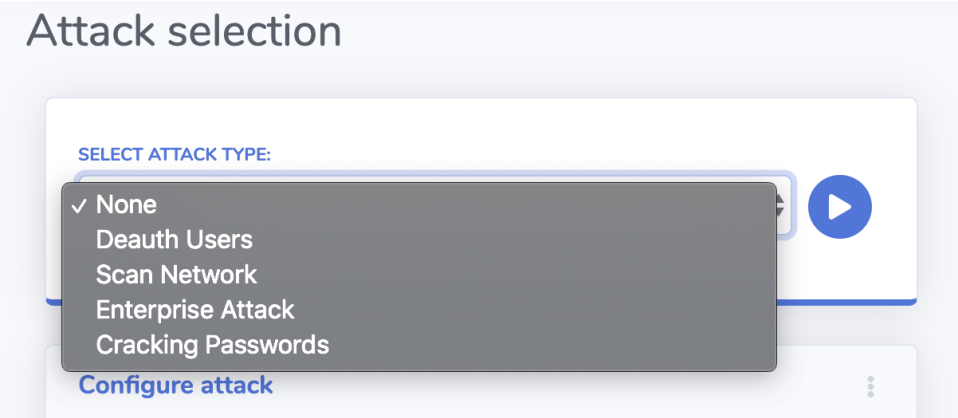


Figura 3.25: Ataques disponibles en WiFi-Pocket

nodejs. Para el caso concreto de las órdenes, realizamos conexiones vía ssh. Estas conexiones hacen uso de un usuario por defecto denominado pi que se crea mediante el script de instalación y la contraseña pi-pocket (esta contraseña es genérica). Los paquetes son interceptados por la aplicación tcpdump y procesados por wifi-pocket para ser posteriormente enviados al panel de control. De la lógica aplicada a los datos, es decir, del procesamiento de los mismos, se encarga el panel de control.

3.7.4. Daemon

Las sondas disponen de su propio fichero de instalación para facilitar al usuario la configuración de WiFi-Pocket. Las opciones de las que dispone el fichero de instalación pueden verse en la figura .

Para poder tener el proceso de las sondas ejecutandose en segundo plano y que este pueda ser arrancado al encender los dispositivos se necesita crear un demonio. La función que nos crea y activa el demonio es la siguiente:

```
createdaemon () {
    pwd=\$(pwd)
    mkdir /etc/systemd/system/wifi-pocket
    echo "[Unit]" > /etc/systemd/system/wifi-pocket/sondes.
        service
    echo "Description=Active probes wifi-pocket service" >> /etc
        /systemd/system/wifi-pocket/sondes.service
    echo "After=network.target" >> /etc/systemd/system/wifi-
        pocket/sondes.service
    echo "StartLimitIntervalSec=0" >> /etc/systemd/system/wifi-
        pocket/sondes.service
    echo -e "\n[Service]" >> /etc/systemd/system/wifi-pocket/
        sondes.service
    echo "Type=simple" >> /etc/systemd/system/wifi-pocket/sondes
        .service
    echo "Restart=always" >> /etc/systemd/system/wifi-pocket/
        sondes.service
    echo "RestartSec=1" >> /etc/systemd/system/wifi-pocket/
```

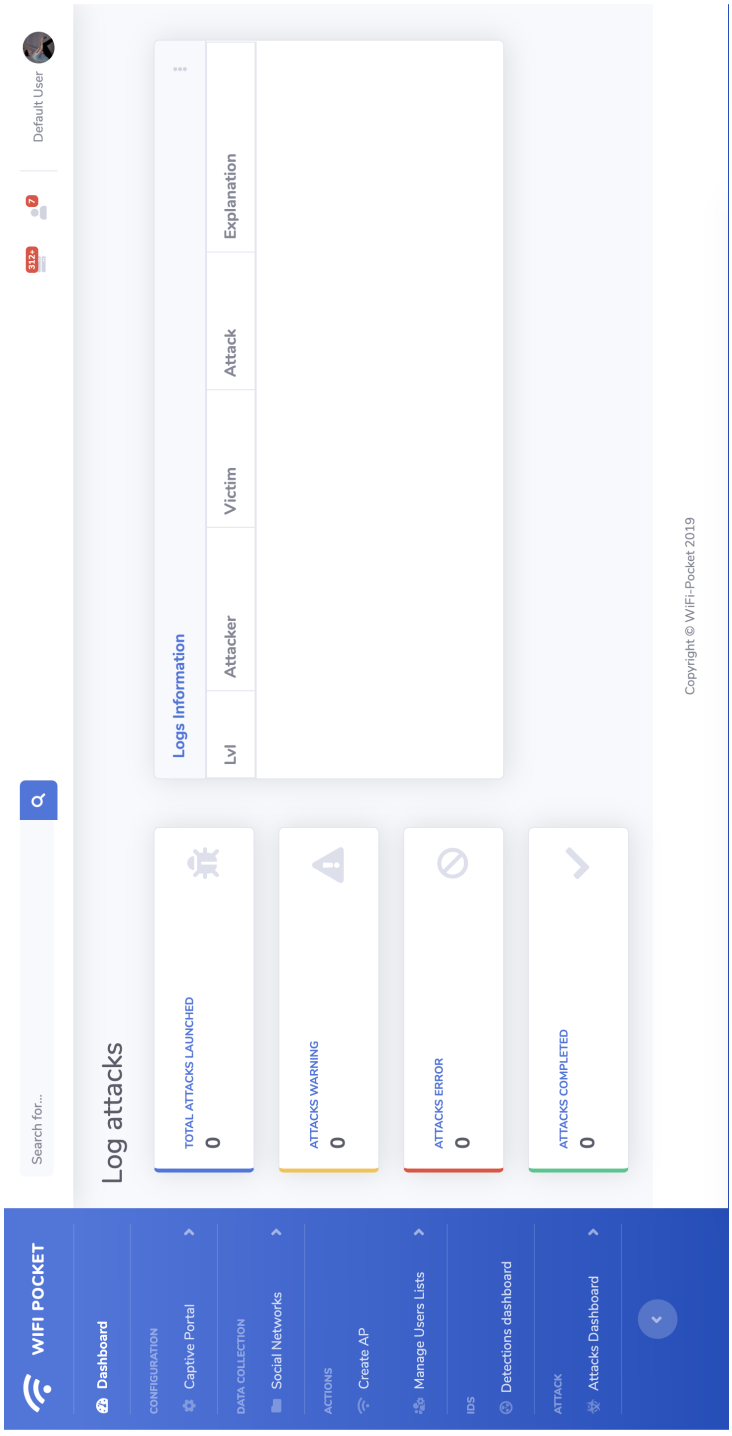


Figura 3.26: Registro con los ataques que se han producido o se están produciendo

Figura 3.27: Sonda del proyecto WiFi-Pocket

```
        sondes.service
    echo "User=\$1" >> /etc/systemd/system/wifi-pocket/sondes.
        service
    echo "ExecStart=/usr/bin/node  \$pwd/main.js" >> /etc/
        systemd/system/wifi-pocket/sondes.service
    echo -e "\n[Install]" >> /etc/systemd/system/wifi-pocket/
        sondes.service
    echo "WantedBy=multi-user.target" >> /etc/systemd/system/
        wifi-pocket/sondes.service

    chmod 777 /etc/systemd/system/wifi-pocket/sondes.service
    systemctl daemon-reload
    systemctl enable /etc/systemd/system/wifi-pocket/sondes.
        service
    systemctl start sondes.service
}
```

A través de este demonio, nos aseguramos que, en caso de cortes de electricidad, la sonda pueda levantarse y seguir con su ejecución.

3.7.5. Sniffando el tráfico

Para interceptar los paquetes, necesitamos hacer uso de un programa como tcpdump (ver 2.7.1.1), y tener una de las interfaces de red en modo monitor (ver 2.4.6.2). La ejecución de la herramienta tcpdump se hace a través de NodeJS. El objetivo es poder analizar cada línea obtenida con tcpdump y darle nuestro propio formato. A través del servidor de sockets levantado, enviaremos esta información obtenida.

Hay que tener en cuenta que los parámetros que le pasamos a tcpdump y que

han sido usados en WiFi-Pocket son por el tipo de paquetes que se necesitaba obtener. En caso de mejoras de la herramienta, es posible que dichos parámetros se vean modificados, ampliando la cantidad de información a recolectar.

En el siguiente fragmento de código se puede observar esta captura de paquetes, parseo de los datos y envío de la información.

```
const WebSocket = require('ws');
const wssProbe = new WebSocket.Server({ port: 4000 });

const parser = require('./parser.js');
const spawn = require('child_process');
const ls = spawn.spawn('tcpdump', ['-G', '2', '-l', '-I', '-
    i', 'wlan1', '-e', '-s', '256', 'type', 'mgt', 'and', '
    subtype', 'deauth', 'or', 'subtype', 'disassoc', 'or', '
    subtype', 'probe-req', 'or', 'subtype', 'beacon']);

var packages = [];

// Broadcast to all.
wssProbe.broadcast = function broadcast(data) {
    wssProbe.clients.forEach(function each(client) {
      if (client.readyState === WebSocket.OPEN) {
        client.send(data);
      }
    });
};

ls.stdout.setEncoding('utf8');
ls.stdout.on('data', (data = data.toString()) => {
    data = data.split(/\r\n|\r|\n/);
    //console.log("\n\n% Tamanio: " + (data.length - 1) + "
       Array: " + data, 'color: #bada55');
    for(var i = 0; i < (data.length - 1); i++){
      if(data[i].search("Probe Request") != -1){
          console.log(parser.PROBRQ_package(data[i].split("
              ")));
          wssProbe.broadcast(parser.PROBRQ_package(data[i].
              split(" ")));
      }
      else if(data[i].search("DeAuthentication") != -1){
          console.log(parser.DEAUTH_package(data[i].split("
              ")));
          wssProbe.broadcast(parser.DEAUTH_package(data[i].
              split(" ")));
      }
      else if(data[i].search("Beacon") != -1){
        console.log(parser.BEACON_package(data[i].split(" ")
            ));
        wssProbe.broadcast(parser.BEACON_package(data[i].
            split(" ")));
    }
    }
});
```

```
    ls.stderr.on('data', (data) => {
      console.error(`stderr: ${data}`);
    });

    ls.on('close', (code) => {
      console.log(`child process exited with code ${code}`);
    });
```

3.7.6. Parseando los datos

La información que nos llega a través de tcpdump tiene su propio formato. Este formato no nos resulta útil para poder analizar la información, por lo que deberemos parsearlo y transformarlo en un formato útil. Este formato útil será el formato interpretado por el panel de control para la detección de posibles ataques y generación de alertas.

Debido a la detección de ataques que se ha implementado, solo nos resultan de utilidad paquetes de typo PROBRQ, DEAUTH y BEACON. En caso de ampliar el rango de ataques a detectar, habría que ampliar el parseador.

Cada tipo de paquete tiene su propia estructura y debe ser tratado de forma diferente. Debido al ataque, de cada paquete tendremos unas necesidades de información y otras.

A continuación se muestra el código que se ha utilizado para parsear la información de los tres tipos de paquetes diferentes.

3.7.6.1. Paquetes PROBRQ

A través de este paquete, podemos saber, por ejemplo, si los usuarios de nuestra infraestructura tienen dispositivos que, por su configuración, pueden considerarse peligrosos al estar generando posibles vectores de ataque.

```
function PROBRQ_package(res){
    return JSON.stringify({
        "type" : "PROBRQ",
        "date" : res[0],
        "signal" : res[6],
        "channel" : res[13],
        "BSSID" : res[14].replace(/SA:/g,''),
        "ESSID" : res[19].replace(/[\(\)]/g,''),
        "client" : res[16]
    });
}
```

Es por ello, que necesitamos saber quien genera la información, y que está preguntando al punto de acceso. De tal forma que podamos detectar al usuario que supone un peligro para nuestra infraestructura, y la razón de dicho peligro.

3.7.6.2. Paquetes DEAUTH

La razón de este paquete viene explicada a través de la sección 2.4.5.2. El objetivo es interceptar estos paquetes para que, desde el panel de control, se controle si

se está produciendo un alto número de paquetes de este estilo que pudiera significar que se está produciendo un ataque.

```
function DEAUTH_package(res){
    return JSON.stringify({
        "type" : "DEAUTH",
        "date" : res[0],
        "signal" : res[10],
        "channel" : res[13],
        "BSSID" : res[14],
        "source" : res[20].replace(/[\(\)]/g,''),
        "destination" : res[17]
    });
}
```

En este caso, necesitamos saber hacia quien va el paquete, y desde donde está siendo emitido, de forma que podamos detectar cual es la persona o red objetivo del ataque.

3.7.6.3. Paquetes BEACON

Este tipo de paquete lo capturamos para detectar nuevos puntos de acceso y poder tener una visión amplia de las redes WiFi que hay alrededor.

```
function BEACON_package(res){
    return JSON.stringify({
        "type" : "BEACON",
        "date" : res[0],
        "signal" : res[6],
        "channel" : res[5],
        "BSSID" : res[12].replace(/BSSID:/g,''),
        "ESSID" : res[20].replace(/[\(\)]/g,''),
        "source" : res[16].replace(/SA:/g,''),
        "destination" : res[15].replace(/DA:/g,'')
    });
}
```

3.7.6.4. Ventaja de JSON

El formato que se usa, aunque nosotros elijamos que campos van a ser enviados, es JSON. Este formato permite gestionar muy bien el intercambio de información e incluso permite introducir la información directamente en la base de datos de MongoDB.

A continuación, se ven los datos que se han introducido en la base de datos.

```
{
    "_id" : ObjectId("5d7538e029f33a0ad9a70a10"),
    "hash" : "JAZZTEL_SqMA40:f4:07:9e:be:b0",
    "attacker" : "none",
    "victim" : "40:f4:07:9e:be:b0",
    "type" : "WARNING",
    "tittle" : "Steal information",
```

```
    "message" : "Someone can create an AP with the ESSID:
        JAZZTEL_SqMA and steal information of 40:f4:07:9e:be:b0"
}
```

En la figura 3.28 se muestra como se ven los datos a través de la aplicación Robo 3T, utilizada para gestionar las colecciones y documentos de MongoDB.

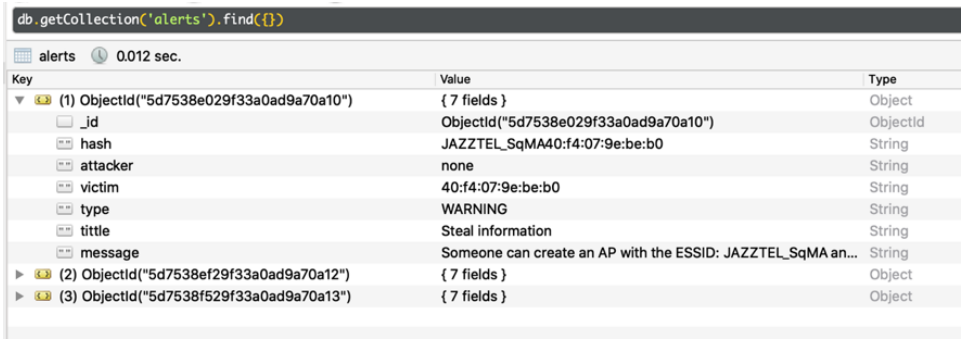


Figura 3.28: Vista de un documento en el programa Robo 3T

3.8. Sistema de Detección de Intrusos - IDS

El Sistema de Detección de Intrusos (IDS) que implementa WiFi-Pocket consta de un servidor de sockets que recoge la información proveniente de las sondas, la procesa mediante unas reglas y genera información a través de un servidor de sockets. Esta información será leída por el navegador web del usuario que consulte el panel de control.

3.8.1. Configuración

Al igual que se facilitan archivos por defecto para la configuración del servidor web, servidor DNS, servidor DHCP, ... También se ha tenido en cuenta facilitar la configuración del núcleo del IDS. Como el lenguaje empleado es NodeJS, las dependencias se instalarán a través de npm.

Para instalar las dependencias, deberemos ejecutar sobre la carpeta cp_sondes el siguiente comando:

```
1    npm install
```

Esto nos creará una carpeta llamada node_modules con todos los módulos que se usan. Estos módulos son ws y mongodb. Para que la configuración sea sencilla, se creó el paquete package.json. Este paquete contiene la siguiente información:

```
{
    "name": "wifi_pocket_control_panel",
    "version": "1.0.0",
    "description": "receive and send data via sockets",
    "main": "main.js",
    "scripts": {
```

```
        "test": "echo \"Error: no test specified\" && exit 1"
    },
    "repository": {
        "type": "git",
        "url": "git+https://github.com/Sawyer13/WiFi-Pocket.git"
    },
    "author": "Sawyer13",
    "license": "GPL-3.0",
    "bugs": {
        "url": "https://github.com/Sawyer13/WiFi-Pocket/issues"
    },
    "homepage": "https://github.com/Sawyer13/WiFi-Pocket#readme"
        ,
    "optionalDependencies": {
        "bufferutil": "^4.0.1",
        "utf-8-validate": "^5.0.2"
    },
    "dependencies": {
        "mongodb": "^3.3.2",
        "ws": "^7.1.2"
    }
}
```

3.8.2. Funcionamiento

En la figura 3.22 veíamos el dashboard del IDS totalmente vacío, sin información. Por defecto, el IDS permanece desactivado. Si un usuario de WiFi-Pocket, desea hacer uso de esta función, deberá pulsar sobre el botón "Start the IDS"situado en la esquina superior derecha.

Como indicabamos en el apartado 3.6.2, las acciones en el panel de control se realizan sin tener que refrescar la página. En cuanto el botón sea presionado, empezará a funcionar.

El funcionamiento será visible cuando en la parte izquierda empiecen a llegar todos los paquetes interceptados por las sondas. Por cada paquete, se realiza un análisis en busca de posibles amenazas. Este análisis generará las alertas que aparecerán en la parte central del panel. En la figura 3.33 podemos ver al IDS en funcionamiento una alerta de precaución y una de peligro generadas.

Distinguimos tres tipos de niveles de alerta. “All ok” de color verde que correspondería al nivel 1, “Warning” de color amarillo que correspondería al nivel 2 y “Danger” de color rojo que correspondería al nivel 3 (ver 3.6.3.5).

El estado de alerta, o nivel de alerta, se encuentra en la zona superior central. Por otro lado, la casilla de alertas se encuentra en la zona inferior central.

Cuando está todo en orden, la casilla de alertas está vacía, sin notificaciones. En el estado de alerta nos encontramos con una carita sonriente que nos dirá que todo está en orden (figura 3.30).

En el momento en el que se produce una alerta, como por ejemplo, cuando detectamos que un dispositivo está preguntando por redes conocidas, principal se pone en el estado de warning (ver figura 3.31). En este momento no hay datos que indiquen que se está produciendo un ataque o que alguien se está aprovechando de este riesgo de seguridad.

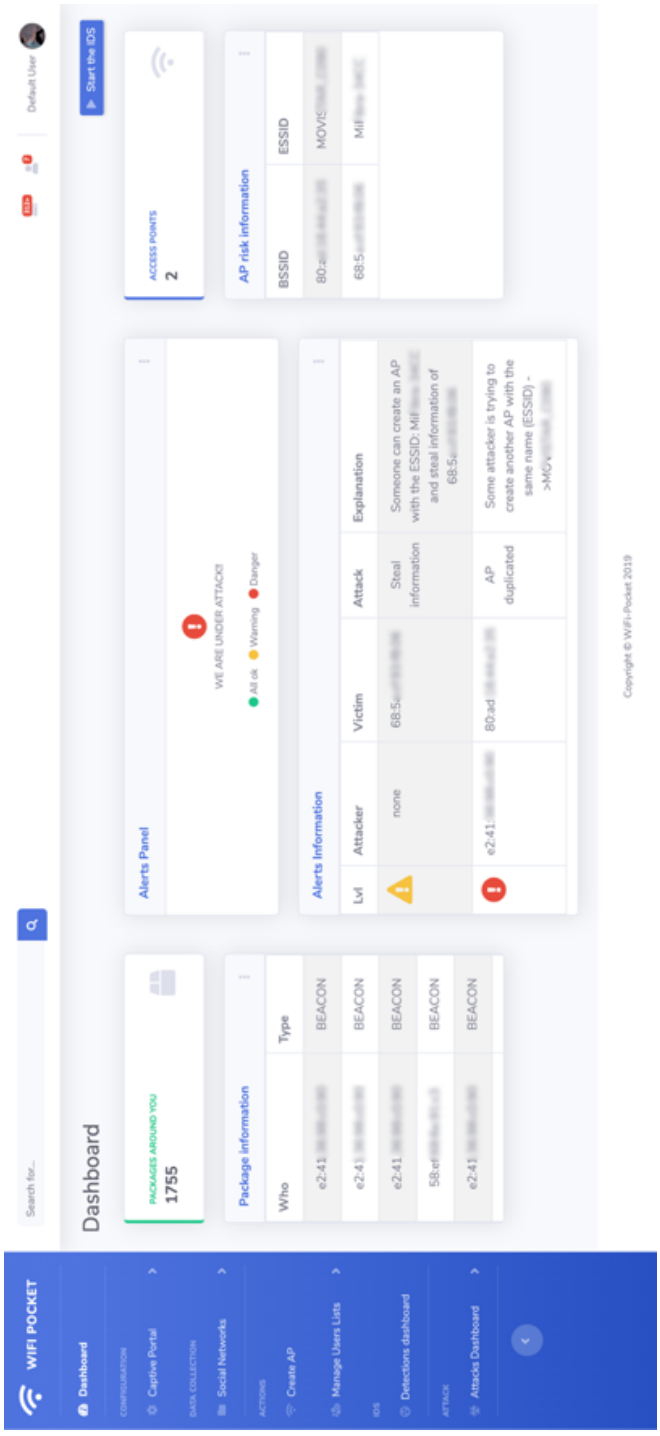


Figura 3.29: Interfaz del IDS activada

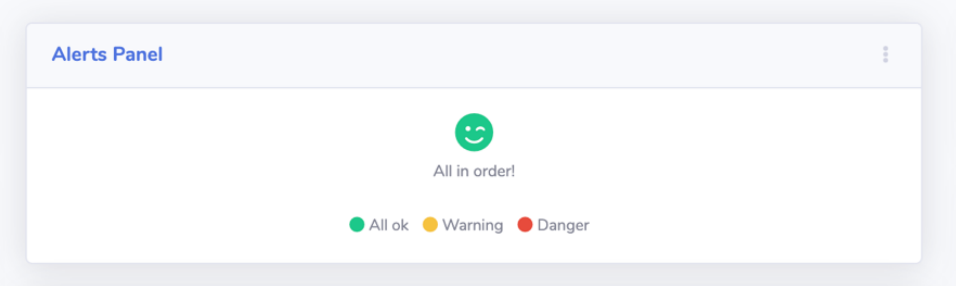


Figura 3.30: Estado de alerta de nivel 1

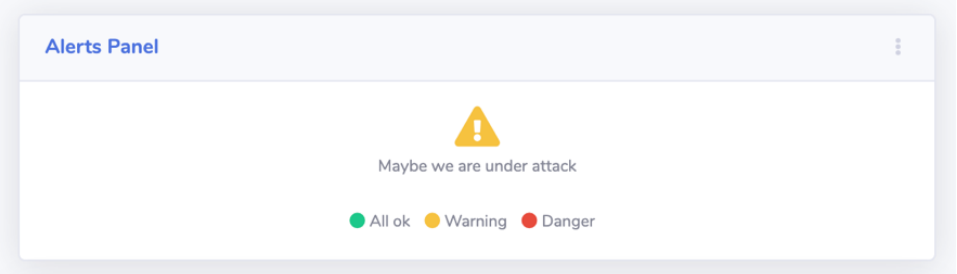


Figura 3.31: Interfaz del IDS activadaS

A la par que se cambia el nivel de alerta, se genera una notificación en el apartado de notificaciones. Esta notificación nos muestra el dispositivo que es vulnerable, el tipo de ataque que se puede producir (robo de información, en este caso) y se acompaña con un comentario aclaratorio acerca de la alerta. Para esta advertencia, el comentario dice que un atacante podría crear un punto de acceso con el nombre “XXXX” y podría robar información al usuario con la dirección MAC "XX:XX:XX:XX:XX:XX".

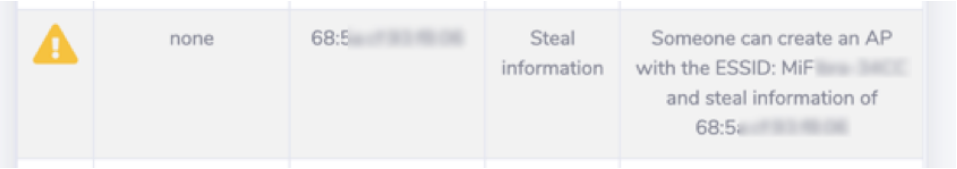


Figura 3.32: Notificación de nivel 2

En el momento que se detecta que se está produciendo un ataque, se genera una alerta y la casilla principal se pone de color rojo, en estado de peligro (ver figura).

En este caso, la notificación nos dice la dirección del atacante, la dirección de la víctima, el tipo de ataque (duplicación de un punto de acceso) y un comentario. El comentario indica que un atacante está tratando de crear otro punto de acceso con mismas características que otro de los puntos de acceso que hay alrededor.

La generación de notificaciones y los cambios del nivel de alerta se genera en tiempo real. Para que esto se produzca, se ha de levantar un servidor de websockets que recogerá la información de las sondas, la procesará, y la enviará a los clientes

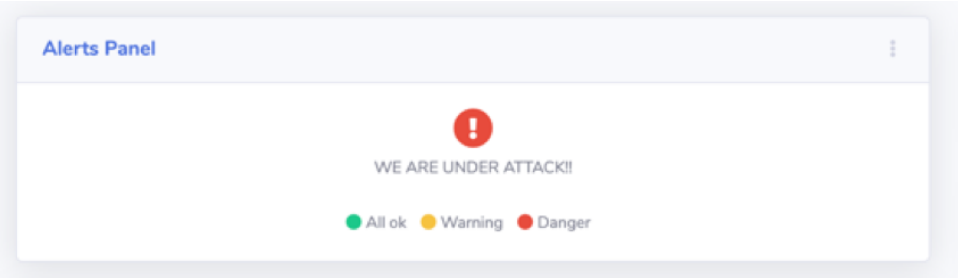


Figura 3.33: Estado de alerta de nivel 3

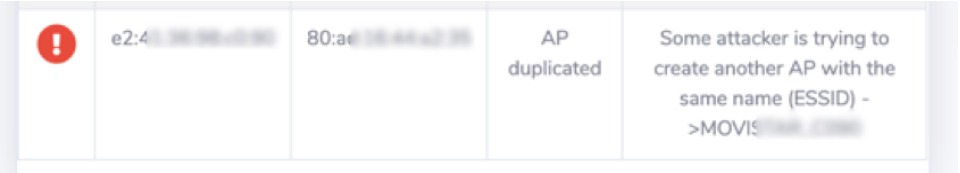


Figura 3.34: Notificación de nivel 3

que escuchan la información.

La creación de dicho servidor se encuentra en el siguiente código. La dirección IP "192.168.100.42" y el puerto "4000" son los datos correspondientes a una de las sondas. En este caso el servidor se genera sobre el puerto 3000.

```
const mongo = require('mongodb').MongoClient;
const url = 'mongodb://localhost:27017';
const WebSocket = require('ws');
const rules = require('./rules.js');

const wssControlPanel = new WebSocket.Server({ port: 3000 });
// Broadcast to all.
wssControlPanel.broadcast = function broadcast(data) {
    wssControlPanel.clients.forEach(function each(client) {
        if (client.readyState === WebSocket.OPEN) {
        client.send(data);
        }
    });
};

mongo.connect(url, {
    useNewUrlParser: true,
    useUnifiedTopology: true
    }, (err, client) => {
    if (err) {
    console.error("err")
    return
    }else{
    // Select the sondes
    (new WebSocket('ws://192.168.100.42:4000', 'echo-protocol'))
        .onmessage = function (
```

```
        evt
    ) {
        //send the package
        wssControlPanel.broadcast(JSON.stringify({'type': '
            package', 'data': JSON.parse(evt.data)}));

        //checkRules
        rules.checkRules(JSON.parse(evt.data), client.db('wifi-
            pocket'), wssControlPanel);
    }
    }
})
```

La información se analiza en la función "checkRules". Esta función se encuentra en el archivo rules.js. A continuación, se muestra un ejemplo de una de las reglas creadas, de tipo "DANGER". La información de las alertas se va depositando en la base de datos.

```
if(data.type == "BEACON"){
    if(data.ESSID != null){
        ap_information.find({ESSID: data.ESSID}).toArray((err,
            items) => {
            if (err) {
                console.log("Error tryint to find ap_information
                    ");
            }else{
                if(items.length > 0){
                    console.log(items);
                    if (items[0].BSSID != data.BSSID){
                        alert_json = {
                            'attacker': data.BSSID,
                            'victim': items[0].BSSID,
                            'type': 'DANGER',
                            'tittle': 'AP duplicated',
                            'message': 'Some attacker is trying
                                to create another AP with the
                                same name (ESSID) ->'+data.ESSID
                        };
                        alerts.insertOne(alert_json, (err,
                            result) => { if(!err) wssControlPanel
                            .broadcast(JSON.stringify({'type': '
                            alert', 'data': alert_json})); });
                    }
                }else{
                    if(data.ESSID != ""){
                        ap_information.insertOne(data, (err,
                            result) => { if (!err)
                            wssControlPanel.broadcast(JSON.
                            stringify({'type': 'ap_data', 'data':
                             data})); });
                    }
                }
            }
        });
```

```
    }
}
```

Para el caso de esta alerta, como el paquete es de tipo "BEACON"m se está comprobando si existe en la base de datos un punto de acceso que contenga el mismo nombre que el que nos viene indicado en el paquete. Si la dirección perteneciente a ese nombre es distinta, significa que se está levantando un punto de acceso con mismo nombre que otro ya existente.

En la subsección 3.7.6 se explica como es el formato a través del cual se recibe la información en el servidor de websockets del panel de control.

3.9. Acciones

En este apartado se mezclan las acciones que se pueden realizar con los ataques que hay disponibles.

A su vez, se pueden combinar varias acciones. Podemos crear un punto de acceso, desautenticar a todos los usuarios y escanear a los objetivos que se conecten a nuestro punto de acceso creado.

Un ejemplo de los scripts que hay implementados en WiFi-Pocket es el siguiente:

```
#!/bin/bash

# Iniciar modo monitor
echo "Iniciando modo monitor en vap mon1 con tarjeta wlan1..."
airmon-ng check kill >/dev/null 2>/dev/null ; sleep 1
airmon-ng check kill ; sleep 1
airmon-ng check kill
iw dev wlan1 interface add mon1 type monitor 2>/dev/null

# Cambiar MAC ambas interfaces
echo "Cambiando MAC en tarjeta WiFi"
ifconfig mon1 down
ifconfig wlan1 down
macchanger -A mon1
macchanger -A wlan1
ifconfig mon1 up

# Iniciar Fake AP (configurar: /etc/hostapd.conf)
echo " Iniciando Fake AP"
killall hostapd ; sleep 2
echo "interface=wlan0" > /etc/hostapd.conf
echo "driver=nl80211" >> /etc/hostapd.conf
# Using name for ssid of ap
ssid="No entrar :)"
echo "ssid=$ssid" >> /etc/hostapd.conf
echo "channel=11" >> /etc/hostapd.conf
echo "hw_mode=g" >> /etc/hostapd.conf
echo "auth_algs=1" >> /etc/hostapd.conf
echo "ctrl_interface=/var/run/hostapd" >> /etc/hostapd.conf

hostapd -B /etc/hostapd.conf
```

```
# Configurar direccion IP
ifconfig wlan0 192.168.0.1 netmask 255.255.255.0 up

# Iniciar servidor DHCP
service dhcpd start

# Configurar firewall
sysctl -w net.ipv4.ip_forward=1
iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
iptables -F FORWARD
iptables -A FORWARD -j ACCEPT
```

Estos scripts se pueden encontrar en la carpeta llamada scripts del repositorio de código. Son scripts sencillos escritos en lenguaje bash. Existe un script de instalación de las dependencias requeridas por dichos scripts. En la figura 3.35 se pueden ver todos los scripts.

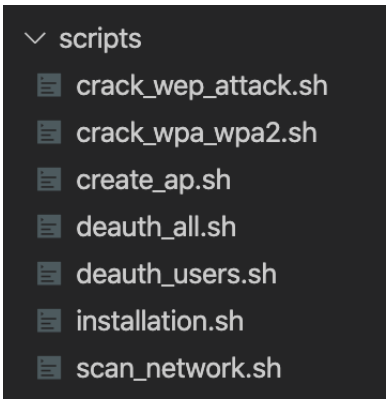


Figura 3.35: Archivos con scripts para Red Team

Siguiendo la categorización en ataques realizada para la vista web, nos encontramos con las siguientes categorías:

- Deauth users: esta categoría recoge los ataques de desautenticación realizados contra los usuarios de una red WiFi. Hace uso de dos de los scripts:

  deauth_user.sh: desautentica a un único usuario.

  deauth_all.sh: desautentica a todos los usuarios de alrededor.

- Scan network: esta categoría recoge la acción de escaneo de red que permite elaborar ataques más elaborados. Se usa el siguiente script:

  scan_network.sh: a través del uso de la herramienta nmap, permite lanzar diferentes tipos de escaneos de red.

- Enterprise attack: este ataque es un ataque concreto que se puede llevar a cabo a través del siguiente script:

  create_ap.sh: crea un punto de acceso haciendo uso de una de las interfaces de red. Resulta de mayor utilidad, para no depender de un servidor freeradius, hacer uso de la herramienta que viene en el script de instalación

"hostapd-wpe" disponible en los repositorios de la distribución de ciberseguridad Kali.

- Cracking passwords: esta categoría recoge los ataques que se realizan contra las medidas de seguridad que emplean los routers. Los scripts que se encuentran dentro de esta categoría son:

  crack_wpe_attack.sh: script automatizado para crackear la seguridad de WPE.

  crack_wpa_wpa2.sh: script automatizado para crackear la seguridad de WPA y WPA2.

3.10. Patrones de Diseño

Un patrón de diseño tiene por objetivo el ahorro de tiempo, la seguridad de ir por un camino allanado (ya recorrido por otros desarrolladores) y permite seguir un modelo sobre el que, en caso de trabajar en equipo, los compañeros de desarrollo pueden seguir y continuar.

Para el presente Trabajo de Fin de Grado se ha considerado de utilizad la aplicación del patrón DAO.

3.10.1. Patrón DAO

El patrón DAO, Data Access Object, busca que la aplicación sea lo más independiente posible de una base de datos o de la fuente de datos a la cual vamos a acceder para extraer o guardar información.

WiFi-Pocket es una herramienta pensada para ser usada por el mayor número de dispositivos de hardware posible. Por eso puede ocurrir que en un determinado momento se prefiera utilizar un modelo de base de datos más ligera o leer los datos directamente desde archivos. Este patrón, nos permite almacenar la información sin saber de dónde se obtiene o dónde va a ser guardada.

La idea es sencilla, tendremos un objeto para cada tipo de dato que contenga sus atributos y funciones set y get, y a la par un objeto DAO para las consultas a la base de datos o la extracción de información.

Así pues, con la implementación del patrón DAO, el sistema de ficheros se muestra en la figura 3.36.

3.11. Bases de Datos

En esta sección veremos los distintos elementos que componen nuestras bases de datos.

3.11.1. Modelo de BBDD Relacional

Para las bases de datos relacionales hemos utilizado el lenguaje SQL. A continuación se muestra el Modelo Entidad Relación, la Estructura de la Base de Datos y la Estructura de las Tablas.

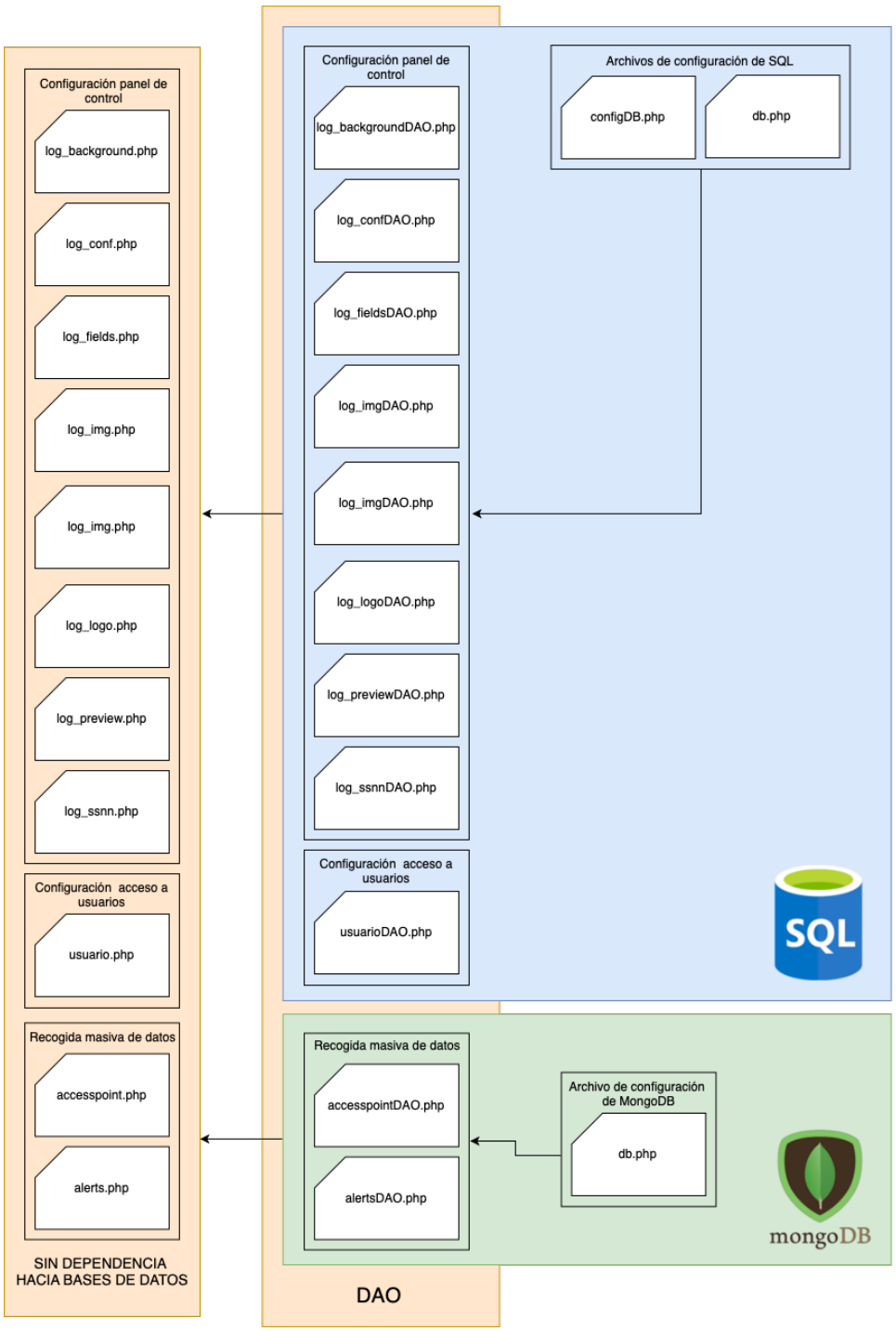


Figura 3.36: Aplicación del patrón DAO en WiFi-Pocket

3.11.1.1. Estructura de la BBDD

A continuación en la figura 3.37 se muestra toda la estructura de la base de datos con sus relaciones, atributos, tablas, claves primarias, claves secundarias, ...

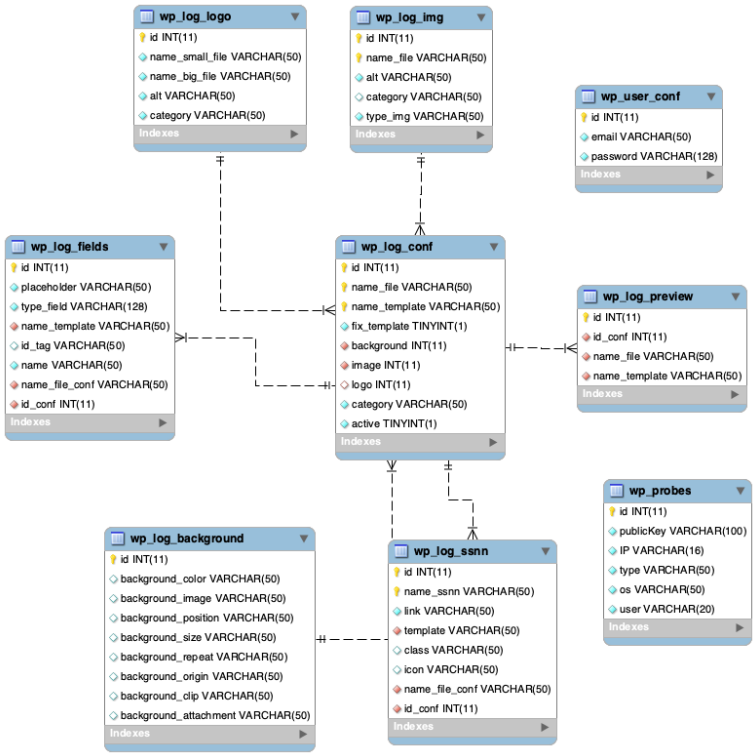


Figura 3.37: Tablas y relaciones de la base de datos relacional

3.11.1.2. Estructura de las Tablas

En esta sección se explica la utilidad de cada tabla creada y los atributos que posee.

- Tabla wp_log_conf: a través de esta tabla guardamos la configuración que tiene el diseño del Captive Portal.
  - id: identificador de la tabla.
  - name_file: nombre del archivo principal que contiene la configuración del portal de acceso.
  - name_template: nombre del diseño utilizado para el portal de acceso.
  - fix_template: marca si el template utilizado es el original o ha sido modificado. Normalmente todos los templates habrán sido modificados.
  - background: hace referencia al id de la tabla wp_log_background.
  - image: hace referencia al id de la tabla wp_log_img.

  - logo: hace referencia al id de la tabla wp_log_logo.
  - category: hace referencia a la categoría del diseño del acceso web.
  - active: solo hay una entrada con valor 1 para indicar cual es la configuración del diseño del portal de acceso en ese momento.

- Tabla wp_log_ssnn: esta tabla permite gestionar las redes sociales que se mostrarán en el portal de acceso.
  - id: identificador de la tabla.
  - name_ssnn: identifica el nombre de la red social.
  - link: identifica el link de la red social.
  - template: hace referencia al atributo name_template que se encuentra en la tabla wp_log_conf.
  - class: guarda la información que usará el objeto que identifique a la red social a utilizar en el portal de acceso.
  - icon: guarda el icono que usará el objeto que identifique a la red social a utilizar en el portal de acceso.
  - name_file_conf: hace referencia al atributo name_file que se encuentra en la tabla wp_log_conf.
  - id_conf: hace referencia al id de la tabla wp_log_cong.

- Tabla wp_log_background: esta tabla permite establecer las características del fondo a visualizar a través del portal de acceso.
  - id: identificador de la tabla.
  - background_color: guarda el valor del atributo en css backgroundcolor.
  - background_image: guarda el valor del atributo en css backgroundimage.
  - background_position: guarda el valor del atributo en css backgroundposition.
  - background_size: guarda el valor del atributo en css backgroundsize.
  - background_repeat: guarda el valor del atributo en css backgroundrepeat.
  - background_origin: guarda el valor del atributo en css backgroundorigin.
  - background_clip: guarda el valor del atributo en css backgroundclip.
  - background_attachment: guarda el valor del atributo en css backgroundattachment.

- Tabla wp_log_fields: esta tabla contiene los campos que queremos que se rellenen en el portal de acceso.
  - id: identificador de la tabla.
  - placeholder: guarda el valor del atributo placeholder del input html utilizado para los formularios.

  - type_fields: guarda el valor del atributo type del input html utilizado para los formularios.
  - name_template: hace referencia al atributo name_template de la tabla wp_log_conf.
  - id_tag: guarda el valor del atributo id del input html utilizado para los formularios.
  - name: guarda el valor del atributo name del input html utilizado para los formularios.
  - name_file_conf: hace referencia al atributo name_file que se encuentra en la tabla wp_log_conf.
  - id_conf: hace referencia al id de la tabla wp_log_conf.

- Tabla wp_log_logo: esta tabla gestiona la información del logo o logos que se pueden incluir en el portal de acceso.
  - id: identificador de la tabla.
  - name_small_file: contiene la ruta a la imagen que se muestra en el apartado de configuración.
  - name_big_template: contiene la ruta a la imagen que se muestra en el panel de inicio de sesión.
  - alt: contiene la descripción de la imagen que se va a usar.
  - category: contiene la categoría a la que pertenece el logo que se va a usar.

- Tabla wp_log_img: esta tabla gestiona la información de la imagen o imágenes que se pueden incluir en el portal de acceso.
  - id: identificador de la tabla.
  - name_file: hace referencia al atributo name_file que se encuentra en la tabla wp_log_conf.
  - alt: contiene la descripción de la imagen que se va a usar.
  - category: contiene la categoría a la que pertenece el logo que se va a usar.
  - type_img: contiene el tipo de imagen que se usará.

- Tabla wp_log_preview: esta tabla guarda la información para una simple visualización de los posibles templates a utilizar en el portal de acceso.
  - id: identificador de la tabla.
  - id_conf: hace referencia al id de la tabla wp_log_conf.
  - name_file: hace referencia al atributo name_file que se encuentra en la tabla wp_log_conf.
  - name_template: hace referencia al atributo name_template de la tabla wp_log_conf.

- Tabla wp_user_conf: esta tabla guarda la información de los usuarios, con sus posibles contraseñas, que pueden acceder al panel de configuración del portal de acceso.

- id: identificador de la tabla.
- email: aloja las direcciones de correo que permiten el acceso a la configuración del panel de control.
- password: aloja las contraseñas hasheadas con sha256 de correo que permiten el acceso a la configuración del panel de control.

- Tabla wp_probes: esta tabla guarda la información de las sondas encargadas de recoger la información de los paquetes WiFi intercambiados a su alrededor.
  - id: identificador de la tabla.
  - publicKey: contiene la clave pública de las sondas que permite la comunicación cifrada bidireccional a través de la VPN.
  - IP: contiene la dirección IP de la sonda que estará recabando información de los paquetes WiFi a su alrededor.
  - type: contiene el tipo de sonda que es. Aquí nos podemos encontrar con Raspberry Pi 1, 2, 3, CuBoX-i, ...
  - os: guarda el valor del tipo de sistema operativo como Raspbian o Debian.
  - user: contiene el valor del usuario a través del cual podremos conectarnos vía ssh.

3.11.2. Modelo de BBDD No Relacional

Para las bases de datos relacionales hemos utilizado el lenguaje MongoDB. En el siguiente apartado se muestra el esquema de la disposición de los documentos, colecciones y base de datos empleada.

3.11.2.1. Estructura de la BBDD

A continuación, en la figura 3.38 se muestra toda la estructura de la base de datos con sus documentos, colecciones, atributos, ...

Para la introducción de los datos, se han establecido reglas a nivel de las bases de datos con el objetivo de evitar duplicados. La razón de esto es debido a la posibilidad que se da de que, al hacer uso de la ejecución en varios hilos, un paquete con información semejante quiera introducirse en la base de datos provocando duplicidad de información.

3.11.2.2. Estructura de los Datos

MongoDB a diferencia de SQL no dispone de tablas. La disposición de las bases de datos con este lenguaje es a través de documentos alojados en colecciones, las cuales se encuentran dentro de la base de datos.

Una aplicación muy útil para la gestión de las bases de datos de MongoDB es Robo 3T. Se trata de una aplicación gratuita con una interfaz muy intuitiva.

La base de datos está compuesta por dos colecciones:

- Colección ap_information: esta colección aloja información acerca de los puntos de acceso situados alrededor de las sondas.

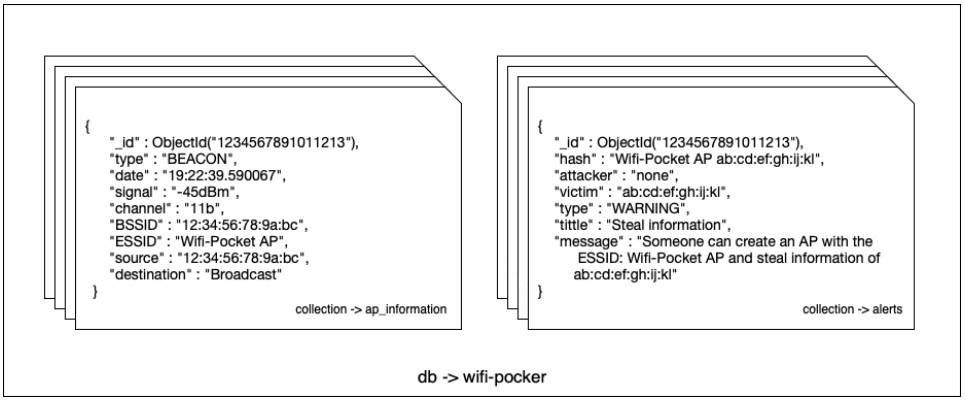


Figura 3.38: Documentos y colecciones de la base de datos no relacional

- id: identificador del documento otorgado por mongodb.
- type: tipo de paquete que ha sido enviado.
- date: fecha en la que se recibió el paquete.
- signal: potencia de emisión del paquete.
- channel: canal sobre el que circula el envío de dicho paquete.
- BSSID: BSSID del punto de acceso emisor del paquete.
- ESSID: BSSID del punto de acceso emisor del paquete.
- source: dirección del emisor del paquete.
- destination: dirección hacia la que ha sido enviado el paquete.

- Colección alerts: esta colección aloja la información de las alertas que se han recibido y que se van recibiendo en tiempo real.
  - id: identificador del documento otorgado por mongodb.
  - hash: hash único que identifica a una alerta. Se busca tener este atributo para evitar la duplicidad de las alertas.
  - attacker: dirección del dispositivo atacante, en caso de haberlo.
  - victim: dirección del dispositivo que está recibiendo el ataque, o es perjudicado por un uso no debido de la red en caso de haberlo.
  - type: tipo de alerta que se envía. Estas alertas pueden ser de dos tipos, WARNING o DANGER.
  - tittle: título corto que posee la alerta.
  - message: descripción completa del significado de la alerta y del posible suceso que está ocurriendo.

3.12. Medidas de seguridad

El presente Trabajo de Fin de Grado está orientado a ser una herramienta que beneficia a la hora de detectar posibles vulnerabilidades y prevenir ataques. eso no

significa que no se haya parado en asegurar la propia herramienta en sí. Todos los datos que se van a recabar tienden a ser privados. Desde recibir credenciales de los usuarios hasta sniffar tráfico en una zona geográfica concreta pudiendo con varias sondas trazar una ruta de movimiento.

Es por este motivo que se vuelve esencial pensar en la ciberseguridad tanto de las comunicaciones que se hagan dentro de WiFi-Pocket, como del almacenamiento de la información. Prevendremos así posibles futuros ataques externos que quieran acceder a nuestros datos o controlar nuestras sondas.

Para poder establecer una política de seguridad adecuada se han utilizado a los siguientes recursos.

3.12.1. Comunicaciones

WiFi-Pocket es una herramienta que aúna el intercambio masivo de información, la interpretación de los datos y la toma de acciones en base a ellos.

La información intercambiada por los servidores de sockets va en texto claro, pudiendo cualquier persona conectarse como cliente a los mismos y recibir dichos datos.

Por otro lado, a veces para la configuración inicial de las sondas necesitaremos conectarnos a las mismas para proceder con el envío de órdenes. Estas órdenes se realizan sobre una terminal con permisos de superusuario.

3.12.1.1. Envío de información

Una de las opciones que se planteó al principio era el intercambio de la información cifrada a través de los propios sockets. Se acabó desestimando por suponer una pérdida de velocidad y por incrementar el número de pasos a realizar para hacer uso de WiFi-Pocket.

Inicialmente, la opción con mejores argumentos a favor fue la de disponer de una red privada virtual, a la que solo pudieran acceder las sondas y el panel de control. De esta forma, toda la información aparecerá como cifrada para personas externas a la red. Nadie podría obtener información de utilidad interceptando nuestros paquetes.

Al hacer uso de una VPN, las conexiones entrantes se reducirían al uso de esa propia red, evitando que los intrusos puedan acceder por otras interfaces de red del propio dispositivo.

La VPN escogida, además, aplica reglas a través de UFW que redirigen todo el tráfico a través de la VPN. Estas reglas, como veremos después, se pueden ampliar para reducir el número de vectores de ataque hacia nuestras sondas.

3.12.1.2. Envío de órdenes

ELas sondas solo son para recopilar información y enviarla al panel de control. Antes de utilizar las sondas como meros chivos espiatorios, tendremos que configurarlas. Además, siempre vendrá bien poder controlarlas de forma remota.

La configuración de las sondas debe realizarse en físico con la sonda delante. La idea es básica, se instala el sistema operativo y se ejecuta el script de instalación de la sonda. Dentro de este script, se permite establecer un usuario concreto para

la sonda con la clave que nosotros designemos. Normalmente, todas las claves de la herramientas serán "pi-pocket" y el usuario será pi.

El objetivo es disponer de un usuario y una contraseña conocidos, para poder establecer una comunicación en remoto con la sonda vía ssh.

El tener una conexión cifrada impide que alguien pueda escuchar el tráfico y pueda visualizar el contenido de nuestras peticiones a la red (que pueda ver, por ejemplo, las páginas que visitamos). También impide que en caso de estar bajo un ataque man in the middle el atacante pueda modificar los paquetes (al estar cifrado un atacante se vería en una gran dificultad para realizar la modificación de los paquetes). Hay otros ataques de los cuales podemos sentirnos seguros haciendo uso de ssh como pueden ser ARP o DNS Spoofing.

3.12.2. Almacenamiento de información

Se ha procurado almacenar la menor cantidad posible información sensible.

Se ha tenido en cuenta para evitar que alguien pueda acceder al panel de control, que la contraseña de los usuarios vaya hasheada. El hash implementado es sha256, un tipo de hash muy fuerte.

Dependiendo de la configuración empleada, la comunicación con el servidor web haciendo uso de Caddy irá cifrada vía HTTPS. Este es un factor que depende de la configuración que se realice de WiFi-Pocket.

3.12.3. Indetección

Al igual que se realiza en la red de la Universidad Complutense de Madrid con sus medidas de seguridad, se busca evitar que cualquier usuario en la red pueda atacar a otros usuarios o pueda escanear al resto de dispositivos, incluyendo las propias sondas.

A pesar de haber establecido unas pocas reglas en iptables para evitar detección vía ping, como trabajo futuro se pretende mejorar los posibles ataques a las sondas y al panel de control.

3.13. Instalación

3.14. Guías

De la preparación para el panel de control. 3.39.

3.15. Scripts

3.15.1. Panel de control

Son scripts en bash con sus propias funciones y menú:

Ejemplo de función:

Menú del script de instalación del panel de control con todas las opciones:

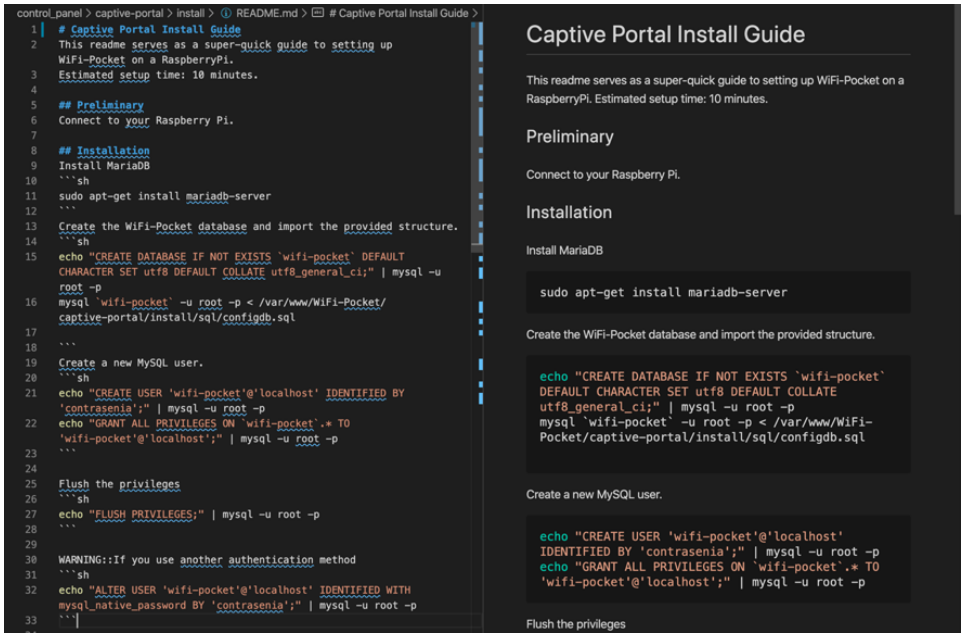


Figura 3.39: Guía para la instalación del panel de control

Figura 3.40: Logo de WiFi-Pocket en los scripts

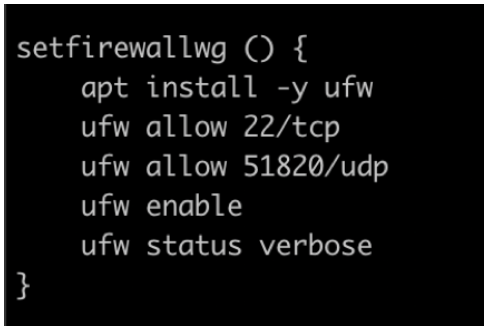


Figura 3.41: Ejemplo de una función en bash

Configuran reglas de iptables, habilitan el firewall, ... E incluso comprueban permisos de super usuario.

Con prevención de posibles errores:

```
updatekeyring () {
    apt update 2>&1 1>/dev/null | sed -ne 's/.*NO_PUBKEY //p' |
        while read key; do if ! [[ ${keys[*]} =~ "$key" ]]; then
        sudo apt-key adv --keyserver hkp://pool.sks-keyservers.
        net:80 --recv-keys "$key"; keys+=("$key"); fi; done
}
```

3.15.2. Sondas

Menú más amplio. La idea es configurarlo, y ya queda todo hecho. Las claves por defecto usadas en los scripts son: Usuario: pi Clave: pi-pocket.

3.16. Documentación

Como se pretende que este Trabajo de Fin de Grado sea continuado y mejorado por otros desarrolladores, se ha puesto empeño en poder dejar documentado el código, añadiendo ficheros de instalación e incluso README's con información acerca del uso y de las dependencias que nos podemos encontrar.

Esta documentación está escrita en inglés, implementando Markdown en los ficheros README.

```
main () {
    logo
    whoami=$(whoami)
    checksudo $whoami

    wifi_networks=$(iw dev |grep Interface |wc -l)

    interfaces $wifi_networks

    while :
    do
        menu

        read answer
        case $answer in
            0)
                exit
            ;;
            1)
                dependencies
            ;;
            2)
                configvpn
            ;;
            3)
                stopvpn
            ;;
            4)
                spanishkeyboard
            ;;
            *)
                echo "Opción no encontrada"
            ;;
        esac

        clear
        logo
    done
}
```

Figura 3.42: Menú del script de instalación del panel de control en código

```
checksudo () {
    if [ $1 != root ];
    then
        echo "WARNING: What are you doing!? Use root user ¬¬"
        exit
    fi
}
```

Figura 3.43: Función que comprueba permisos de superusuario

```
read answer
case $answer in
    0)
        exit
    ;;
    1)
        dependencies
    ;;
    2)
        createdaemon $whoami
    ;;
    3)
        newpiuser
    ;;
    4)
        launchservice
    ;;
    5)
        systemctl enable ssh
        systemctl start ssh
    ;;
    6)
        configvpn
    ;;
    7)
        stopvpn
    ;;
    8)
        killprocess
    ;;
    9)
        spanishkeyboard
    ;;
    *)
        echo "Opción no encontrada"
    ;;
esac
```

Figura 3.44: Menú del script de instalación de las sondas en código

Wifi Pocket

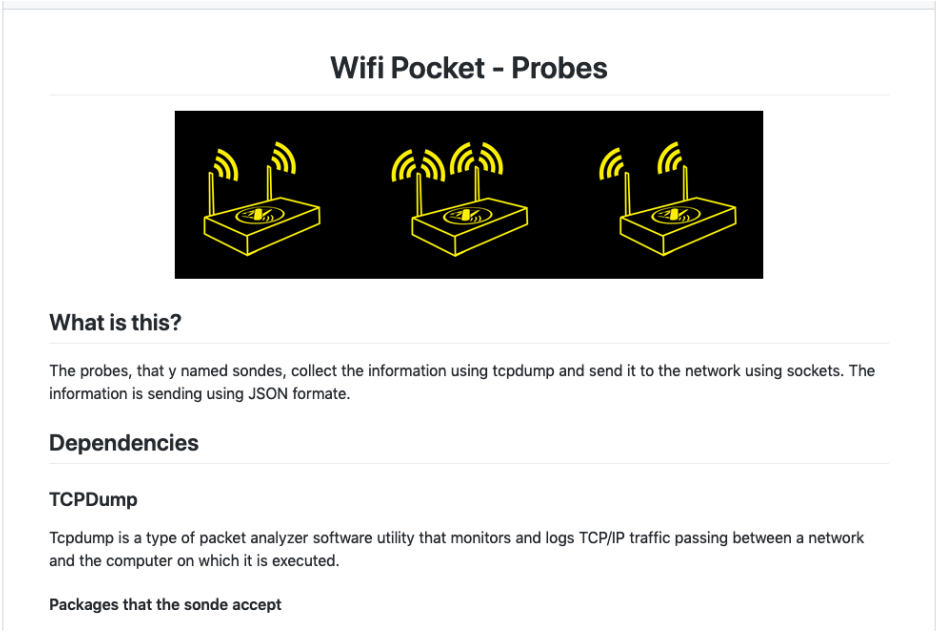
Made with Bash License GPLv3 awesome hacking hardware toolkit

About Wifi-Pocket

Wifi-Pocket wants to be a suite for Red and Blue teams. The idea is simple:

- ¿Red Team? No problem, you have a control panel with many simple attacks to do.
- ¿Blue Team? No problem, use the probes and control the activity with the rules and the control panel.

Figura 3.45: Fichero de documentación principal README.md



Wifi Pocket - Probes

What is this?

The probes, that y named sondes, collect the information using tcpdump and send it to the network using sockets. The information is sending using JSON formate.

Dependencies

TCPDump

Tcpdump is a type of packet analyzer software utility that monitors and logs TCP/IP traffic passing between a network and the computer on which it is executed.

Packages that the sonde accept

Figura 3.46: Fichero de documentación de las sondas README.md

Capítulo 4

Tecnología Empleada

"Ahora pueden construir lo que quieran; la www es una plataforma para que ustedes construyan lo que imaginen."
— Tim Berners-Lee

4.1. Lenguajes Back-End

4.1.1. NodeJS

NodeJS es un entorno de ejecución para JavaScript, multiplataforma de código abierto con licencia MIT, para la capa del servidor basado en el lenguaje de programación ECMAScript y en el motor V8 de Google.

Si algo atrae de Node es su funcionamiento asíncrono, que utiliza un hilo de ejecución principal para la aplicación, del que se generan el resto de las llamadas concurrentes de entrada/salida. Se trata de una arquitectura orientada a eventos.

Otro de sus atractivos es la alta capacidad de la comunidad colaborativa que rodea a Node. Este entorno posee módulos que pueden añadir una simple funcionalidad como un servidor por websockets. Esto permite que estemos ante un entorno muy escalable, pudiendo aplicar no solo el módulo mencionado, sino multitud de ellos.

Su parte asíncrona, no bloqueante, que permite tener diferentes tareas en ejecución de forma simultánea, le convierte en el lenguaje perfecto para manejar sondas que estén enviando información de forma continuada para la realización de tareas en tiempo real basada en dicha información.

Este lenguaje se ha implementado en las sondas para poder lanzar en segundo plano un sniffer de tráfico, parsear en tiempo real los paquetes interceptados y enviar la información al panel de control de WiFi-Pocket. Se ha comprobado que gracias a este lenguaje no se pierde ningún paquete.

4.1.2. PHP

PHP es un lenguaje de código abierto bajo licencia PHP que está destinado a ejecutarse en el lado del servidor. Fue diseñado para el preprocesado de texto

plano en UTF-8. Se debe tener en cuenta que PHP no genera HTML, sino que es compatible con este. Esto nos permite poder hacer uso de HTML en nuestros documentos de PHP.

Podemos programar nuestros scripts en lenguaje PHP para manipular la salida de HTML que enviaremos al usuario, sin que el usuario sepa qué manipulación hemos hecho.

Actualmente nos podemos encontrar con múltiples funciones desarrolladas en PHP que nos facilitarán la tarea de programar y nos ahorrarán tiempo.

Este lenguaje de programación está muy bien documentado en su web principal. Esto le convierte en el lenguaje más recomendable a usar por su extrema simplicidad para el principiante. También nos ofrece muchas características avanzadas para los programadores profesionales.

A través de este lenguaje se ha podido crear la parte back-end del panel de control de WiFi-Pocket, que permite ejecutar tareas en la parte del servidor y realizar las conexiones con las bases de datos.

4.2. Lenguages Front-End

4.2.1. HTML y CSS

HTML (1999-2020) es un lenguaje de marcado muy utilizado para la creación de páginas web desarrollado por Tim Berners Lee. Actualmente es un estándar al cargo de la World Wide Web Consortium.

Su funcionamiento es a través de tags. Podemos encontrarnos con tags para texto, imágenes, tablas, links, ... La función de maquetar y construir todo el contenido enviado recae sobre el navegador.

Por otro lado, CSS (1999-2020) es un lenguaje de hojas de estilo. Su objetivo es modificar el aspecto de los documentos HTML. Actualmente todas las páginas web hacen uso de CSS para mostrar su contenido. Una de sus características clave es permitirnos poder visualizar el texto en diferentes tamaños a través de diferentes dispositivos electrónicos.

El código y diseño del panel web del captive portal y del panel de control ha sido moldeado a través de HTML5 y CSS3. Se ha podido crear un portal responsive y accesible a través de un navegador web desde cualquier dispositivo.

4.2.2. JavaScript

Este lenguaje de programación interpretado, JavaScript (1999-2020), se utiliza principalmente en la parte del cliente permitiendo a los sitios y aplicaciones web mejoras en la interfaz de usuarios y un mayor dinamismo.

Uno de sus puntos fuertes es que permite relegar una gran carga de las operaciones desde el servidor al cliente, agilizando procesos y dando una sensación de mayor velocidad y rendimiento.

Si se combina con otras tecnologías como AJAX, este lenguaje de programación nos permite también poder establecer una comunicación vía cliente-servidor en tiempo real.

Actualmente, la mayoría de sitios y aplicaciones web utilizan este lenguaje de

programación.

En el caso de WiFi-Pocket, se ha podido crear una página más dinámica. A su vez, el motor del IDS ha sido creado con este lenguaje. Recibimos la información a través de websockets y la procesamos mediante unas reglas construidas en JavaScript.

4.2.3. Bootstrap

Actualmente, la mayoría de sitios y aplicaciones web hacen uso de frameworks que facilitan su construcción y funcionamiento. Dentro de los frameworks más famosos de diseño web se encuentra Bootstrap, una biblioteca multiplataforma de código abierto bajo licencia MIT.

Este Framework contiene una serie de diseños predefinidos, como plantillas de diseño con tipografías, formularios, botones, cuadros, menús de navegación, ..., que facilitan la maquetación de una web.

Algunas de las clases que vienen por defecto en este Framework han sido modificadas para ser adaptadas a WiFi-Pocket y a las necesidades visuales que han ido surgiendo. Es una librería que ha facilitado la creación del portal cautivo y del panel de control.

4.2.4. jQuery y AJAX

Hoy en día nos encontramos con multitud de librerías que nos facilitan la tarea de elaboración de código y que nos permite, por lo tanto, ahorrar tiempo. Entre algunas de esas librerías, por ejemplo, esta jQuery.

jQuery es una librería multiplataforma de código abierto bajo la licencia MIT y GPL que facilita la interacción de manera muy simple con el árbol DOM de las plantillas HTML asociadas a un script. Nos simplifica mucho la gestión de tareas como la adición de texto sobre un campo HTML, la modificación de las clases del documento, la modificación del estilo, ... También, nos ayuda en la tarea de conexión direccional con el servidor a través de AJAX.

AJAX (1999-2020) (Asynchronous JavaScript And XML), por otro lado, nos permite crear aplicaciones web interactivas. La comunicación se realiza desde el cliente, mientras en segundo plano a través de AJAX se va realizando una comunicación asíncrona con el servidor. El contenido no tiene por que estar en XML, podemos enviar contenido, por ejemplo, a través de JSON.

Con el fin de darle más dinamismo a la configuración del captive portal a través del panel de control, se ha implementado tanto jQuery como AJAX en WiFi-Pocket. Podemos ir moldeando el portal cautivo a la par que seguimos añadiendo modificaciones. Estas modificaciones no se ven en tiempo real sobre el captive portal sino que se ven al momento sobre la propia vista de configuración del captive portal.

4.3. Otras tecnologías

4.3.1. JSON

Wikipedia (JSON), JavaScript Object Notation, es la alternativa actual al intercambio de datos a través de XML. A través de este formato de texto sencillo,

nos resultará más fácil escribir un parseador (analizador sintáctico) para el mismo.

Aunque en su nombre aparezca JavaScript, no es necesariamente parte de JavaScript.

Actualmente es usado en muchos sistemas que requieren mostrar o enviar información para ser interpretada por otros sistemas, independientemente del lenguaje de programación utilizado. Podemos comunicarnos haciendo uso de JSON por ejemplo, entre sockets creados con NodeJS y sockets creados con Python.

La información que intercambia WiFi-Pocket con las sondas es haciendo uso de JSON. Este formato permite escribir un parseador más cómodo, que ayuda a traducir los datos interceptados a través del sniffer y a darles un formato que sea más fácil de interpretar por el panel de control.

4.3.2. Markdown

Markdown es un lenguaje de marcado ligero bajo licencia BSD. El objeto es poder dotar al usuario de un lenguaje en texto plano fácil de leer con posibilidad de ser convertido a HTML.

En el caso de WiFi-Pocket, Markdown ha sido utilizado en la documentación para facilitar su lectura y mejorar la calidad gráfica de la misma. En las comunidades de software libre, se valora mucho la documentación.

4.3.3. Shell Scripting

Cuando hablamos de Shell Scripting estamos hablando de la creación de un programa o script que será ejecutado a través de la shell de Unix. Existen varios lenguajes de scripting, como Bash.

Este lenguaje se ha empleado en la automatización de tareas y en la elaboración de scripts de WiFi-Pocket. Un ejemplo de ello son todas las sondas que cuentan con un script inicial de configuración que permite configurar paso a paso todos los servicios de los que requiere una sonda. A través de este script podemos configurar la VPN, habilitar las interfaces de red, establecer el fichero principal como demonio que se ejecute al arranque previniendo posibles fallos por perdidas de corriente, detectar posibles fallos causados por el hardware, ...

4.3.3.1. Bash

Bash es un lenguaje de scripting escrito en C bajo licencia GPLv3. Su función consiste en interpretar órdenes. Actualmente es el intérprete de comandos usado en las distribuciones GNU/Linux y en el sistema operativo macOS X. Es posible hacer uso de bash en sistemas operativos como Android o Windows.

Dentro de los lenguajes de Shell Scripting, es Bash el que se ha empleado en Wifi-Pocket.

4.3.4. Python

Actualmente nos encontramos con que los lenguajes predominantes en el mundo de la programación son C y JAVA. En el año 2020, como se muestra en la figura 4.1,

el lenguaje de programación Python se convirtió en el tercer lenguaje mundialmente más usado, y pretende ser el lenguaje predominante.

Jan 2020	Jan 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.896%	-0.01%
2	2		C	15.773%	+2.44%
3	3		Python	9.704%	+1.41%
4	4		C++	5.574%	-2.58%
5	7	^	C#	5.349%	+2.07%
6	5	v	Visual Basic .NET	5.287%	-1.17%
7	6	v	JavaScript	2.451%	-0.85%
8	8		PHP	2.405%	-0.28%
9	15	^^	Swift	1.795%	+0.61%
10	9	v	SQL	1.504%	-0.77%
11	18	^^	Ruby	1.063%	-0.03%
12	17	^^	Delphi/Object Pascal	0.997%	-0.10%
13	10	v	Objective-C	0.929%	-0.85%
14	16	^	Go	0.900%	-0.22%
15	14	v	Assembly language	0.877%	-0.32%
16	20	^^	Visual Basic	0.831%	-0.20%
17	25	^^	D	0.825%	+0.25%
18	12	vv	R	0.808%	-0.52%
19	13	vv	Perl	0.746%	-0.48%

Figura 4.1: Top 20 lenguajes de programación

Se trata de un lenguaje de programación interpretado, cuyo objetivo primordial es favorecer un código legible.

Nos encontramos en la sintaxis de Python, que no se hace uso como en otros lenguajes estilo C y JAVA, con la falta de "{ }", entre otras cosas.

Este lenguaje es un lenguaje multiparadigma, es decir, soporta programación imperativa, funcional e incluso programación orientada a objetos.

Python (2017) s un lenguaje multiplataforma, bajo la licencia Python Software Foundation License.

La desventaja que se puede encontrar en el uso de este lenguaje es su dependencia del intérprete. A veces nos encontraremos con que al ejecutarse procesos hijos dependientes del padre, estos estarán constantemente reconectando con el proceso padre sin ser en su totalidad un proceso independiente.

Se ha investigado su uso probando el lenguaje en las sondas pero ha sido descartado. Pese a ello, debido a que la mayoría de herramientas de ciberseguridad ofensivas para redes WiFi están construidas en este lenguaje, se baraja su posible implementación en un futuro como lenguaje de scripting.

4.4. Bases de Datos

Llamamos bases de datos a los conjuntos de datos que son almacenados sistemáticamente y pertenecen a un mismo contexto.

Hoy en día, la mayoría de las bases de datos se encuentran en formato digital. Dentro de este formato, nos encontraremos con diferentes categorizaciones.

4.4.1. Relacionales

Las bases de datos relacionales cumplen con el modelo basado en la lógica de predicados y la teoría de conjuntos, es decir, con el modelo relacional.

Su idea fundamental se basa en el uso de relaciones, permitiendo a las herramientas garantizar la no duplicidad de registros, la integridad referencial llegando incluso a poder eliminar todos los registros dependientes tan solo eliminando uno.

Actualmente todos los sistemas de bases de datos relacionales hacen uso de SQL (Structured Query Language).

4.4.1.1. MariaDB

MariaDB es un fork realizado sobre MySQL que nos aporta un mayor rendimiento y nuevas funcionalidades. Por esta razón, cuando se habla de MariaDB podríamos decir que estamos hablando de MySQL.

Se trata de un servicio de manejo de bases de datos relacionales bajo la licencia GPL creado por el propio desarrollador de MySQL junto a un grupo de desarrolladores que decidieron formar parte de forma voluntaria. Esta creación se llevó a cabo ante el temor de Oracle pudiera comenzar a distribuir su software MySQL bajo una licencia de pago. A su vez, otra de las razones fue el temor del deterioro de la herramienta al no estar ya en manos de la comunidad.

Algunas distribuciones populares ya traen por defecto MariaDB. Este gestor de bases de datos es utilizado por las grandes compañías como Google, Wikipedia o Mozilla.

Sin duda, al hacer uso en el presente Trabajo de Fin de Grado del sistema operativo GNU/Linux y debido a la potencia de este gestor, se trata de una elección perfecta. Se ha implementado este tipo de bases de datos relacional sobre la infraestructura del panel de control que requería del alojamiento de información con relaciones, como, por ejemplo, la configuración del captive portal.

4.4.2. No relacionales

Cuando hablamos de bases de datos no relacionales estamos hablando de bases de datos NoSQL. Estas bases de datos dejan a un lado el lenguaje SQL para la gestión.

Para la gestión del almacenamiento de los datos no se usan estructuras fijas como tablas y escalan bien horizontalmente.

Este tipo de base de datos surge tras la necesidad de poder procesar grandes cantidades de datos que tenían estructuras horizontales similares.

Al ser no relacionales, no poseen relaciones. Si bien hay maneras de conseguir relaciones, la idea en este tipo de base de datos es tener las menos posibles.

4.4.2.1. MongoDB

MongoDB fue creada por MongoDB Inc. Esta base de datos escrita en el lenguaje de programación C++ esta orientada a documentos, de esquema libre. Dicho de otra forma, nos encontramos con que cada entrada puede tener un esquema de datos nada parecido al resto de datos almacenados.

Destaca por su velocidad a la hora de ejecutar sus operaciones debido al manejo de datos binarios.

Estamos ante una de las bases de datos no relacionales más elegida por los desarrolladores.

La forma que tiene de almacenar la información es a través de un sistema propio de documento conocido con el nombre BSON. Este tipo de documento es una evolución del formato JSON añadiéndole la peculiaridad de poder almacenar datos representados de forma binaria.

Podemos encontrar mongodb en los repositorios oficiales de las distribuciones de GNU/Linux principales.

Este tipo de bases de datos no relacional ha sido implementado en WiFi-Pocket para recoger de forma masiva toda la información que se obtiene, por ejemplo, de los puntos de acceso. Esta información no presenta relaciones.

4.5. Servidor Web

Un servidor web, también conocido como servidor HTTP, es un servidor virtual, no físico. Nos permite procesar una aplicación en el lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente.

El código recibido por el cliente es renderizado por un navegador web.

4.5.1. Caddy

Caddy es un servidor web escrito en Go bajo licencia Apache 2. Fue creado en el año 2014 pero no se publicó hasta el año 2015. Actualmente es un servidor web poco conocido.

Actualmente este servidor web es más seguro que otros como Apache. Destaca sobre todo por sus módulos y la automatización de tareas, a la vez que por su sencillez de configuración.

Esta configuración se basa en un archivo principal denominado CaddyFile, a través del cual podemos establecer los dominios de los que hará uso nuestro servidor junto a los módulos o configuraciones (versión de PHP, compresión bajo gzip, ...).

Un módulo a destacar es:

- http.git: este módulo nos permite tener un repositorio donde, cada vez que hagamos push de forma externa, recibiremos la notificación a través de un hook previamente configurado en clientes como GitHub actualizando automáticamente el contenido en nuestro servidor.

  Nos permite elegir qué rama queremos que sea la visible en un dominio configurado para la visualización de forma externa. Normalmente podremos tener una rama main con la versión estable que se actualizará automáticamente si se producen cambios.

También existen módulos que nos permiten realizar accesos bajo autenticación de manera sencilla e incluso módulos que nos permiten controlar los intentos de acceso a nuestro servidor, bloqueando a aquellas direcciones ip atacantes.

Si queremos hacer uso de certificados para mejorar la seguridad utilizando peticiones HTTPS, podemos hacer uso de Let's Encrypt, que nos permite obtener certificados totalmente gratuitos. Esta implementación viene por defecto en el servidor web y no requiere de un script o plugin adicional.

Este servidor es recomendable si vamos a volcar toda la lógica de la aplicación a servidores externos, como en el prototipo vamos a hacerlo sobre local, no se ha empleado este servidor.

4.5.2. Apache

Apache HTTP Server es un servidor web que implementa el protocolo HTTP. Se trata de un servidor web gratuito y de código abierto bajo licencia Apache 2.0.

Cuando un cliente solicita un recurso al servidor, este se le es ofrecido de vuelta. Toda la información enviada al cliente incluirá todos los recursos necesarios para su visualización.

Apache aporta bases de datos de autenticación y negociación de contenido. Carece de una interfaz gráfica que ayude en su configuración.

Muchos programadores de aplicaciones web utilizan una versión local de Apache con el fin de previsualizar y probar código mientras este es desarrollado. En nuestro caso, nuestro servidor web no estará disponible a través de la red de Internet.

Este servidor web no dispone por defecto de automatización para la generación de certificados digitales gratuitos.

Apache ha sido el servidor web que se ha implementado sobre el prototipo. Este servidor web es el más utilizado en la actualidad y no posee una configuración compleja. WiFi-Pocket dota al usuario que va a configurar la herramienta de archivos por defecto de configuración de Apache.

4.5.3. NodeJS Server

El lenguaje de programación NodeJS nos permite a través de módulos como express levantar servidores que pueden ser de tipo web. Estos servidores actúan de modo similar a Caddy o Apache, con el único inconveniente de no permitirnos gestionar direcciones de dominio.

Como vimos anteriormente, el servidor caddy posee un fichero denominado Caddyfile y el servidor Apache posee los virtual hosts. La única forma viable de tener nuestro panel central escrito en NodeJS sería haciendo uso de otro servidor web que gestionase los dominios y redireccionase las peticiones a través de un proxy hacia el servidor NodeJS levantado. Para esto, necesitaríamos levantar tantos servidores NodeJS como dominios tengamos, con su correspondiente uso de puertos.

El servidor de NodeJS se ha implementado en el intercambio de información a través de los websockets. Cada sonda levanta su propio servidor. Se ha descartado su uso en el servidor web que aloja el panel de control por requerir este de la gestión de nombres de dominio.

4.6. Tunel ssh

El protocolo SSH (secure shell) se utiliza para "tunelizar"tráfico confidencial sobre Internet de una manera segura.

Por ejemplo, un servidor de ficheros puede compartir archivos usando el protocolo SMB (Server Message Block), cuyos datos no viajan cifrados. Esto permitiría que una tercera parte, que tuviera acceso a la conexión (algo posible si las comunicaciones se realizan en Internet) pudiera examinar a conciencia el contenido de cada fichero trasmitido.

Por lo tanto, para poder montar el sistema de archivo de forma segura, se establece una conexión mediante un túnel SSH que encamina todo el tráfico SMB al servidor de archivos dentro de una conexión cifrada SSH. Aunque el protocolo SMB sigue siendo inseguro, al viajar dentro de una conexión cifrada se impide el acceso al mismo.

En este trabajo, para conectarnos con las sondas de forma segura, utilizando SSH, haríamos que el panel de control se conecte a un cliente SSH. El cliente SSH se conectaría con el servidor tunelizado.

Es decir, ¿cuál es la idea? La idea es realizar una conexión que vaya cifrada de forma que aunque esta sea interceptada, no se pueda visualizar su contenido.

4.7. VPN

Una conexión por VPN crea una red privada entre dispositivos aunque estos no estén conectados directamente de forma física entre ellos. Pueden ser ordenadores que se encuentren en distintas partes del mundo. Esta conexión se realiza haciendo uso de Internet.

En cuanto nos conectamos al servidor VPN nuestro ordenador pasará a formar parte de la red privada virtual asemejándose la misma a una red local con todas las características de este tipo de redes.

Imaginemos que queremos conectarnos a la red privada de nuestra Universidad o empresa. Podemos realizar esta conexión a través de una VPN.

La conexión establece un túnel cifrado entre nuestro ordenador y la VPN, de esta forma aunque se intercepte el tráfico no se podrá saber qué uso estamos haciendo. Para poder realizar la conexión ambas partes deberán autenticarse previamente.

Al conectarnos a través de un túnel cifrado nuestro ISP (Proveedor de Servicios en Internet) nunca podrá tener acceso al contenido de nuestra comunicación. Normalmente todas las páginas que consultamos son registradas por nuestro proveedor de servicios en internet, quien, al menos por ley, debe conservarlas durante varios años.

Toda conexión que realizamos con la VPN registra Logs. Dichos logs en caso de ser guardados pasan a disposición de quien nos ofrezca el servicio VPN (podemos ser incluso nosotros mismos quienes creemos nuestra propia VPN).

Al igual que nos pasa con los proxys, nunca debemos de fiarnos de aquellos que ofrecen un servicio gratuito pues pueden estar vendiendo toda la información que recopilan de nuestras conexiones.

Hay proveedores de VPN que aseguran que cifran los LOGS y legalmente permiten a cualquier cuerpo de seguridad que los requiera poder acceder a ellos. El

inconveniente es que al estar cifrados dichos logs no sirven de nada.

Existen proveedores VPN que permiten pagar con tarjetas regalo de amazon, con varias cryptomonedas y diferentes métodos de pago.

Una VPN puede actuar en diferentes niveles del modelo OSI de red.

4.7.1. OpenVPN

OpenVPN es un producto de software libre publicado bajo licencia GPL que nos permite realizar conexiones de punto a punto con validación jerárquica de usuarios y host conectados remotamente.

Una de sus ventajas es su amplia configuración.

Esta VPN implementa conexiones de la capa 2 de enlace o 3 de la capa de red. Usa los estándares del protocolo SSL/TLS para cifrar.

Se ha investigado y probado su uso en WiFi-Pocket. En el prototipo con espacio ideal se ha preferido hacer uso de Wire Guard. No se descarta la implementación de este tipo de VPN en dispositivos hardware como los primeros modelos de Raspberry Pi.

4.7.2. Wire Guard

Es una VPN tan fácil de instalar como ssh. Nos encontramos ante una nueva forma de establecer conexiones privadas que nos permite una mayor eficiencia y fluidez del tráfico, pudiendo alcanzar mayores velocidades de conexión en un hardware más austero. Se trata de un modelo cliente-servidor.

Esta VPN se desarrolló con la idea de integrarse en el Kernel de Linux. Esta integración permite mayor flexibilidad en la configuración de redes, simplificando el proceso de configuración y gestión de los servicios de la VPN.

Podemos encontrar WireGuard integrado en un router con OpenWRT.

Uno de sus problemas es que solo acepta UDP.

Este tipo de VPN solo se aplica sobre el nivel de la capa 3 de red.

En el caso de la Raspberry Pi 2 la instalación puede hacerse de manera automatizada. Para las versiones anteriores y para la Raspberry Pi Zero se debe hacer la instalación mediante configuración manual.

Wire Guard es la VPN escogida para el prototipo con espacio ideal. Es perfecta para el hardware escogido y presenta una configuración sencilla. Al tener scripts de configuración, podemos montar la VPN sin encontrarnos con problemas.

4.8. Sockets

Llamamos socket al método de comunicación entre un programa que representa la parte cliente y un programa que representa la parte del servidor. Se trata de un concepto abstracto a través del cual dos programas establecen un flujo de datos de manera fiable y ordenada.

El mayor uso que se le da a los sockets es a través de un sistema de peticiones llamado interfaz de programación de aplicación de sockets (API, application programming interface).

Para poder establecer una comunicación basada en sockets, es necesario que un programa sea capaz de localizar a otro y establecer conexión con él. A partir de ahí, ambos programas deberán ser capaces de intercambiar información.

El concepto de socket se origina a partir de dos recursos. Por un par de direcciones IP (cliente-servidor) y un par de números de puerto, y queda definido en el caso de utilizar el protocolo TCP/IP.

La comunicación siempre es iniciada a través del cliente. La parte del servidor estará esperando a recibir algún dato para poder empezar con el intercambio de información.

Se puede hacer uso del protocolo TCP o UDP. El protocolo TCP es un protocolo orientado a conexión, por lo que garantiza que no haya errores ni se pierdan paquetes por el camino y que se mantenga un orden en el envío. Por otro lado, el protocolo UDP no está orientado a conexión, con lo que no se garantiza la entrega. Los mensajes en caso de llegar no tienen por qué llegar ordenados.

4.8.1. WebSockets

Aunque parezca que los sockets y los websockets son bastante parecidos debido a su similar funcionamiento, en realidad son bastante diferentes. Los websockets comunmente son ejecutados a través de un navegador que se conecta al servidor de sockets mediante un protocolo similar a HTTP que se ejecuta sobre TCP/IP.

Están destinados principalmente a aplicaciones web que requieren una conexión permanente con el servidor. Por otro lado, los sockets (no websockets) son más potentes y genéricos. Pueden ejecutarse también sobre TCP/IP pero no están restringidos a los navegadores o al protocolo HTTP.

En caso de usar una VPN como Wire Guard cuyos paquetes son enviados a través de UDP, es recomendable usar estos websockets sobre el protocolo TCP/IP asegurándonos la llegada o el reenvío de paquetes, pese a ser segmentos TCP a través de datagramas UDP.

Para el intercambio de información entre las sondas y el panel de control, se levanta un servidor en cada sonda. Este servidor permite el intercambio a través de websockets. Cada vez que se recibe un paquete, se parsea la información y se envía al panel de control.

4.9. DNS

El protocolo DNS, por sus siglas Domain Name System (Sistema de Nombres de Dominio en español), nos permite traducir nombres de dominio (como garcia-baameiro.com), a sirecciones IP. Dicho de otra forma, cuando un usuario teclea un nombre de dominio en su navegador, el servidor DNS le indica cuál es la dirección IP del servidor web que tiene ese nombre de dominio y en el que se puede consultar la información web buscada.

Todas las páginas web en la internet normal están hospedadas bajo direcciones IP.

Un servidor DNS posee una base de datos distribuida y jerárquica que guarda la relación entre los diferentes nombres de dominio y su correspondiente dirección IP.

Como usuarios, podemos configurar los servidores DNS sobre los que queremos que nuestros dispositivos hagan sus peticiones. Sin embargo, a menudo vienen impuestos por defecto, o son especificados a través de protocolos como DHCP. Los principales servidores DNS más conocidos son:

- Google
  - 8.8.8.8
  - 8.8.4.4
- Cloudfare
  - 1.1.1.1
  - 1.0.0.1

Actualmente los servidores DNS de Cloudfare son los que poseen mayor velocidad, y prometen el borrado de datos de las consultas pasadas 24 horas.

Algunas distribuciones de GNU/Linux, como Parrot Security, traen por defecto sus propios DNS. Esto evita el uso de servidores DNS que recibimos a través de DHCP.

Las empresas suelen configurar sus propios servidores DNS para poder tener servicios internos que no estén expuestos a la red de Internet.

WiFi-Pocket presenta archivos de configuración para servidores DNS por defecto que pueden ser usados a modo de referencia en una configuración inicial.

4.10. DHCP

DHCP, por sus siglas Dynamic Host Configuration Protocol (protocolo de configuración dinámica de host en español), es un protocolo de red que realiza la asignación de direcciones IP y otros parámetros de configuración en una red permitiendo la comunicación con otros dispositivos.

Un servidor DHCP dispone de un archivo de configuración que contiene una lista de direcciones IP dinámicas que se va asignando a los clientes según su disponibilidad.

Tanto la asignación de direcciones IP como demás datos de red se realiza de forma automatizada.

Algunos de estos parámetros configurables, son:

- Dirección del servidor DNS
- Nombre DNS
- Puerta de enlace de la dirección IP
- Dirección de Broadcast
- Máscara de subred
- MTU

Para poder asignar a un cliente los datos de red, ha de ser el cliente quien los solicite. La forma de solicitarlo es a través del comando:

```
1    sudo dhclient -r <interfaz>
```

El funcionamiento del protoclo sigue el orden descrito a continuación:

- DHCPDISCOVER: el cliente envía un mensaje de broadcast para descubrir los servidores DHCP disponibles.
- DHCPOFFER: los servidores DHCP de la red realizan una oferta al cliente con los parámetros de configuración. El cliente puede recibir más de una oferta.
- DHCPREQUEST: se realiza una petición de oferta por parte del cliente de tipo broadcast para informar al resto de servidores (también se usa para extender el tiempo de cesión de los datos). Esta petición especifica el nombre del servidor del que finalmente se acepta la oferta.
- DHCPACK: es el mensaje de confirmación y cierre enviado desde el servidor hacia el cliente con los parámetros definitivos (broadcast).
- DHCPRELEASE: el cliente envía un mensaje al servidor para informar de que ya no va a usar más la dirección IP.

En caso de haber un único servidor DHCP en la red, los parámetros de configuración de red recibidos serían de dicho servidor.

La parametros de configuración que un servidor DHCP envía al cliente se establecen en el archivo /etc/dhcpd.conf. En la figura 4.2 se muestra un ejemplo de un archivo de configuración de un servidor DHCP. A través del parámetro "option domain-name-servers" establecemos las direcciones IP de los servidores DNS sobre los que queremos que los clientes realicen sus peticiones DNS.

```
default-lease-time 600;         # 10 minutes
max-lease-time 7200;            # 2  hours

option domain-name "dns.garciabaameiro.com";
option domain-name-servers 192.168.1.1, 192.168.1.2;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;

subnet 192.168.1.0 netmask 255.255.255.0
 {
  range 192.168.1.10 192.168.1.20;
  range 192.168.1.100 192.168.1.200;
 }
```

Figura 4.2: Archivo de configuración /etc/dhcpd.conf

WiFi-Pocket presenta archivos de configuración para servidores DHCP por defecto que pueden ser usados a modo de referencia en una configuración inicial.

Capítulo 5

Estudio del hardware a emplear

"La libertad no es poder elegir entre unas pocas opciones impuestas, sino tener el control de tu propia vida. La libertad no es elegir quien será tu amo, es no tener amo."
— Richard Stallman

5.1. Microprocesadores analizados

En esta sección se habla de todo el hardware que se ha ido probando barajando su posible implementación en el proyecto WiFi-Pocket.

5.1.1. Raspberry Pi

Cuando hablamos de una Raspberry pi hablamos de un mini ordenador de placa reducida, ordenador de placa única u ordenador de placa simple de bajo coste y pequeño tamaño. Esta placa del tamaño de una tarjeta de crédito es uno de los dispositivos IoT más conocidos. Se desarrolló en Reino Unido por la Fundación Raspberry Pi (Universidad de Cambridge). Posee un diseño de hardware libre. Se pueden adquirir en china "copias" que cumplen los mismos requisitos que la Raspberry a un precio inferior. Nació con el objetivos de fomentar la enseñanza de la informática en la educación de las escuelas. Empezó a comercializarse en el año 2012 tras haberse creado en el año 2011.

Como se puede observar en la figura 5.1, se trata de una placa base simple, sin recursos externos como pantalla, teclado o ratón. Por supuesto, presenta los componentes suficientes para poder conectar estos recursos externos a la misma.

Este pequeño ordenador no posee disco duro. Dispone de un lector/ranura para memorias SD, un sistema de almacenamiento en estado sólido. Una de la problemática de las tarjetas SD es la de no poder leer y escribir al mismo tiempo.

Posee un procesador central (CPU) de tipo ARM. La fundación Raspberry da soporte para las descargas de sistemas operativos para la arquitectura ARM, como son:

- Raspbian

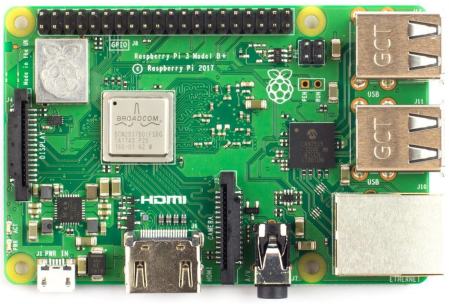


Figura 5.1: Visualización de una Raspberry Pi

- Ubuntu Core
- Ubuntu Server
- Windows 10 IoT Core
- OSMC
- LibreELEC
- Mozilla WebThings
- PiNet
- RISC OS
- Weather Station
- IchigoJam RPi

En nuestro caso haremos uso de Raspbian, la distribución Debian para dispositivos Raspberry con procesador ARM. Esta distribución es la más conocida.

En la página oficial se ofrece una imagen denominada NOOBs, que nos permitirá instalar nuestro sistema operativo.

Dentro del sistema operativo Raspbian, podremos elegir si lo queremos con interfaz gráfica o incluso si queremos que venga con recursos útiles. Esto dependerá de nuestras necesidades de almacenamiento. En el caso de las sondas no se requiere interfaz gráfica, pero en el caso del panel de control sí.

Se debe descargar la imagen (archivo .iso) de la página inicial. Deberemos comprobar también que la imagen no haya sido modificada mediante el uso de los algoritmos de función resumen sha1sum o md5sum. Este último, a pesar de seguir siendo usado no se considera fiable, por lo que se recomienda probar también el segundo.

5.1.1.1. Raspberry Pi 1 modelo A (producto no a la venta)

Actualmente descontinuado, la Raspberry Pi 1 modelo A es el primer modelo de Raspberry. Comenzó a comercializarse en el año 2012. Esta versión no posee puerto Ethernet, obligando al usuario a conectarse a través del puerto USB adaptadores o antenas Wifi.

Comenzó con 26 conectores GPIO. Poseía salida de vídeo a través de HDMI y Video RCA. Para poder conectar altavoces o auriculares, disponía de un conector Jack de 3.5 milímetros. Para la conexión a través de USB solo existía un único conector. Estaba alimentada a través de MicroUSB, necesitando 5 voltios y 2 amperios para su funcionamiento. Esta primera versión disponía de un conector de cámara. El tamaño de la memoria RAM era de 256 MB.

En cuanto al procesador, venía equipado con un Broadcom BCM2835 de un solo core a 700MHz.

El precio inicial fue de 40 euros.

5.1.1.2. Raspberry Pi 1 modelo B (producto no a la venta) y B+

El siguiente modelo, el modelo B, fue también comercializado en el año 2012. Se trata de una variante del modelo A, que incluye algunas mejoras.

Esta variante, disponía del doble de memoria RAM, alcanzando el medio GigaByte. Fue en esta nueva versión cuando por fin se dispuso de un conector Ethernet (RJ-45). A su vez, se amplió en 1 el número de USBs, llegando a ser 2.

El tamaño y el coste de la placa no se vieron modificados, al igual que el procesador y la parte gráfica.

Un tiempo después se mejoraron las prestaciones con el lanzamiento del modelo B+. Esta vez tendríamos 4 puertos USB y en lugar de usar una tarjeta SD se optó por el uso de una MicroSD.

En el presente Trabajo de Fin de Grado, una de las sondas es una Raspberry Pi 1 modelo B. Esta placa tiene dos puertos USB posicionados en vertical, lo que dificulta conectar dos interfaces WiFi a través de USB. Se recomienda utilizar adaptadores WiFi mini USB, como se ve en la figura 5.7.

A su vez, otra de las sondas es la Raspberry Pi 1 Modelo B+.

5.1.1.3. Raspberry Pi 2 modelo B

Dos años después del primer lanzamiento, en el año 2014, se lanza la Raspberry Pi 2 modelo B.

En esta nueva versión, notamos un cambio en el procesador, que pasará de contar con tan solo un núcleo a contar con 4 núcleos. La velocidad aumenta de los 700MHz a los 900MHz.

También, se dobla la cantidad de memoria RAM, llegando ya a tener 1 GB. Al estar compartida con la memoria gráfica el tamaño de la propia placa varia siendo algo inferior.

Los pines aumentan en 14, teniendo 40 pines GPIO.

No ha cambiado nada en cuanto a la interfaz gráfica, el VideoCore IV y las 4 entradas por USB.

Se suprime la conexión RCA.

Se ha implementado esta versión para el presente Trabajo de Fin de Grado.

5.1.1.4. Raspberry Pi 3 modelo B

Dos años después, en el año 2016, contamos con la Raspberry Pi 3 modelo B.

En esta variante, se mejora la velocidad del procesador de 900MHz, manteniendo los cores, a 1.20GHz.

Esta nueva versión destaca sobre todo por la incorporación de una tarjeta WiFi y Bluetooth, permitiéndonos ahorrar en adaptadores.

5.1.1.5. Raspberry Pi 3 modelo B+

Dos años después, tan solo cambiando el modelo, apareció la Raspberry Pi 3 B+ en marzo del 2018.

Este nuevo cambio supuso una mejora en el procesador pasando de tener 1.2Ghz a 1.4Ghz. Incorpora en la conectividad inalámbrica doble banda, a 2,4GHz y 5GHz. La velocidad del Ethernet se triplica, adaptándose a las nuevas ofertas de las compañías telefónicas.

Se actualiza la placa de Bluetooth a la versión 4.2.

5.1.1.6. Raspberry Pi 3 modelo A+

Meses después, en noviembre de 2018, surge el modelo A+. Esta versión redujo en prestaciones disminuyendo el tamaño de la memoria RAM, el número de puertos USB y quitando la conexión de red por cable Ethernet (RJ-45).

5.1.1.7. Raspberry Pi Zero

Tras haber intentado reducir en prestaciones la versión 3 modelo A+, surgió un nuevo tipo de modelo. Estos modelos se considerarían mini ordenadores bajo el nombre de Raspberry Pi Zero.

Notamos como el tamaño es mucho menor, provocando que sean menos potentes.

El precio se reduce drásticamente alcanzando en el mercado los 5-10 euros.

No se ha probado el software en este placa al no disponer de forma nativa de puertos USB ni de módulo WiFi. Se presenta en esta memoria para mostrar la existencia de este hardwre.

5.1.1.8. Raspberry Pi 4 modelo B

En Junio de 2019, surgió la Raspberry Pi 4 modelo B.

Esta nueva versión aporta un pequeño cambio en la reducción del tamaño de los puertos HDMI a dos puertos microHDMI. Este cambio es beneficioso a la hora de manejar dos pantallas de 4K a 60Hz.

Otro de los cambios sustanciales es la mejora de los puertos USB, pasando de ser 2.0 a 3.0.

La eficiencia del procesador se triplica y la velocidad del puerto Ethernet mejora.

Este nuevo modelo ha tenido problemas de sobrecalentamiento en comparación con las versiones anteriores.

5.1.2. GL.iNet

GL.iNet es una empresa china desarrolladora de dispositivos electrónicos. La mayoría de sus productos son routers WiFI con el sistema operativo OpenWRT preinstalado. Estos dispositivos vienen con una versión modificada de este sistema operativo, que incluye software preinstalado útil para el usuario. Este software permite utilizar sus dispositivos como repetidores WiFi y disponer de una red VPN con OpenVPN a través de conexiones con su dispositivo.

Sus productos son bastante económicos y pueden encontrarse en la mayoría de las tiendas de venta online, como Amazon y Aliexpress. El precio no varía mucho de uno a otro, y la compra se realiza a través de su tienda oficial.

Se puede ver una comparativa de los dos productos presentados en el presente Trabajo de Fin de Grado con sus diferentes variables en la figura 5.2.

5.1.2.1. GL-AR150

El producto de GL.iNet GL-AR150 es un mini router ideal para utilizar junto a dispositivos IoT. Dispone de dos versiones, una con antena interna y otra con antena externa. Podemos verlo en la figura 5.3.

Este dispositivo se usa para la construcción de los dispositivos WiFi Pineapple de Hack5.

Sus prestaciones son:

- Powered by Atheros 9331 SoC, 400MHz CPU
- 150Mbps high speed
- 16MB Flash, 64M RAM
- Stable performance
- Small, light, easy to use
- Low power consumption
- OpenWrt pre-installed
- 4 GPIOs for IoT development
- External antenna optional support
- Optional POE support

Sobre todo destaca por su chip Atheros. Actualmente se comercializan con el chip Atheros AR9271 que ha demostrado ser el chip más potente del mercado.

Su tamaño es pequeño y pesa muy poco, lo que facilita que pueda ser llevado en un bolsillo.

Se han hecho pruebas con este dispositivo con OpenWRT y una batería externa, de las que se suelen dar como promoción, de capacidad de 1000mAh. Estas pruebas han demostrado que el dispositivo podría permanecer encendido extrayendo tráfico durante 1 semana sin apagarse.

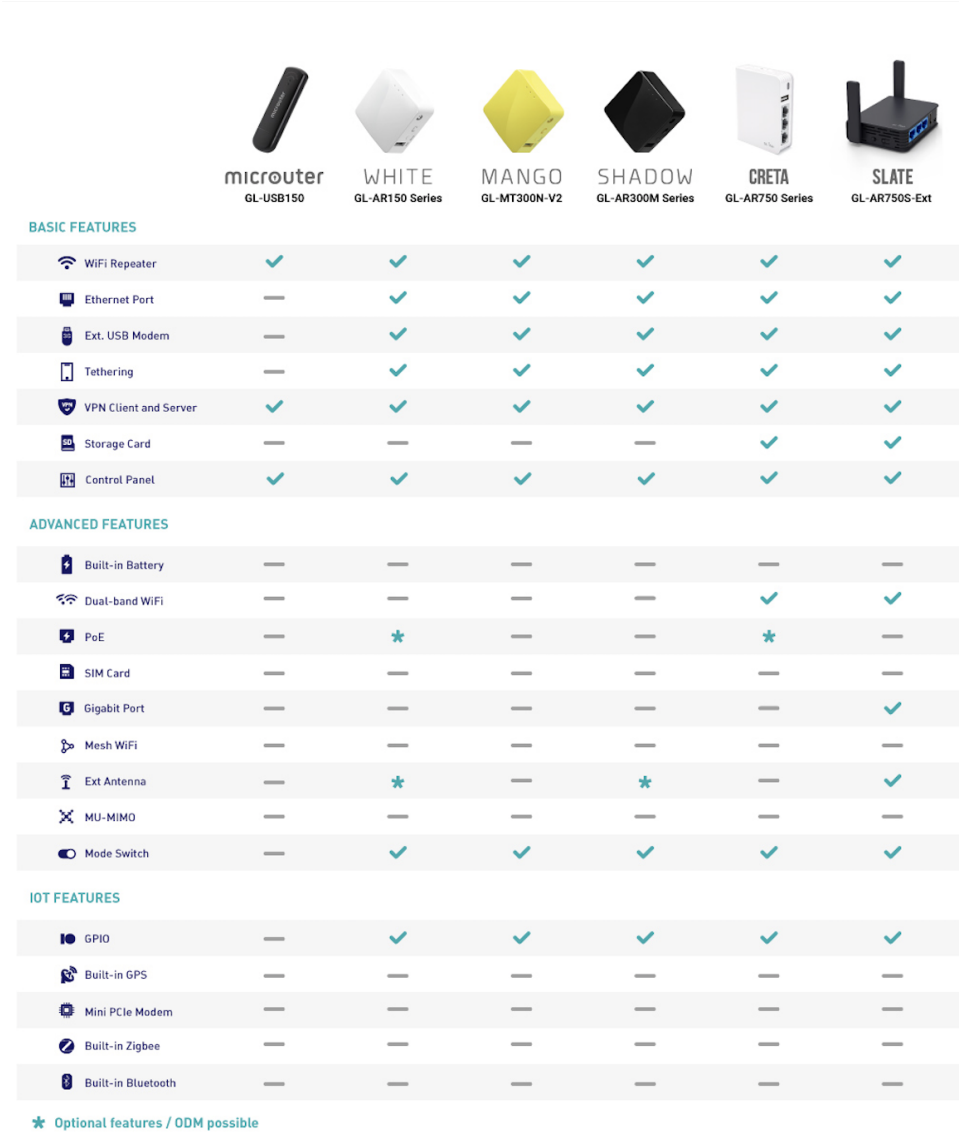


Figura 5.2: Comparativa de los dispositivos GL.iNet

Uno de sus principales inconvenientes es la poca capacidad de almacenamiento (16MB) que posee. Esta memoria FLASH puede ser ampliada modificando el hardware.

El precio ronda los 20-25 euros.

5.1.2.2. GL-AR750

El producto de GL.iNet GL-AR750 está considerado como un router de viaje que podemos guardarnos en el bolsillo. Viene con OpenWRT preisntalado y las

Figura 5.3: Dispositivo GL.iNet GL-AR150

opciones que ofrece GL.iNet como fabricante. Podemos verlo en la figura 5.4.

Sus prestaciones son:

- Powered by Qualcomm QCA9531 SoC,650MHz CPU
- 00Mbps(2.4G) + 433Mbps(5G) high speed
- DDR2 128MB RAM
- Support external storage up to 128GB
- 16MB Nor flash
- Small, light, easy to use
- OpenWrt/LEDE pre-installed

El chip para conexión de red vía WiFi es distinto en comparación con el GL-AR150. Disponemos de la posibilidad de configuración de dos redes, que nos permitirán por un lado esnifar el tráfico y, por otro lado, realizar conexiones externas a través de internet o levantar un punto de acceso para controlar un panel de control.

El consumo que realiza también es bajo. Para el presente TFG no se ha podido probar su consumo en cuanto a duración en tiempo.

El precio es algo más elevado, rondando los 40-50 euros. Puede ser adquirido también en los principales portales de compra online vía web.

El espacio FLASH es de 16MB pero dispone de una ampliación vía tarjeta SD de hasta 128GB.

Este dispositivo no ha sido implementado en la infraestructura de WiFi-Pocket, pero se presenta como una alternativa viable al tener un tamaño pequeño, disponer de dos redes WiFi que permitirían recopilar información por un lado y enviarla por otro, y por su poco consumo de energía. Los únicos defectos vienen dados por su complejidad de configuración y por su alto precio.

Figura 5.4: Dispositivo GL.iNet GL-AR750

Figura 5.5: Dispositivo CuBoX-i

5.1.3. CuBox-i

Los dispositivos CuBox y CuBox-i están desarrollados por la empresa israelí SolidRun. Se trata de una serie de mini ordenadores similares a la Raspberry Pi pero más compacto, de tamaño 5 x 5 x 5cm. Podemos verlo en la figura 5.5

Fueron desarrollados en el año 2011 pero no fue hasta 2012 cuando se comenzaron a comercializar.

A través de su página web oficial se pueden descargar las imágenes de diferentes sistemas operativos. Estas imágenes (la mayoría desactualizadas) no tienen comprobación de función resumen, por lo que no se puede comprobar la veracidad de la descarga.

A diferencia de la Raspberry Pi, las versiones de CuBox no se cargan a través de microUSB. Puede aguantar temperaturas de hasta 70º C y el consumo de electricidad es bajo.

Todos los modelos disponen de la misma estructura hardware. Se diferencian en el número de Cores y el tamaño de la memoria RAM.

La figura 5.6 muestra una comparativa entre los diferentes CuBox-i.

	From $80	From $90	From $100	From $120
Model	CuBox i1	CuBox i2	CuBox i2eX	CuBox i4P
Carrier Type	Pro	Pro	Pro	Pro
System On Chip	i.MX6 Solo	i.MX6 Dual Lite	i.MX6 Dual	i.MX6 Quad
Core Count	1	2	2	4
Memory Size	512MB	1GB	1GB	2GB
Memory Config	32 bit @ 800Mbps	64 bit @ 800Mbps	64 bit @ 1066Mbps	64 bit @ 1066Mbps
GPU	GC880	GC2000	GC2000	GC2000
3D GPU Type	OpenGL ES1.1,2.0	OpenGL ES1.1,2.0	OpenGL ES1.1,2.0 Quad Shader	OpenGL ES1.1,2.0 Quad Shader
Accelerated Media Enc/Dec	See Below	See Below	See Below	See Below
HDMI 1080p with CEC	1.4, 3D support	1.4, 3D support	1.4, 3D support	1.4, 3D support
WiFi 11n	Optional	Optional	Optional	Built In
BlueTooth	Optional	Optional	Optional	Built In
Powered USB 2.0	2	2	2	2
Ethernet	10/100/1000 Mbps (*)	10/100/1000 Mbps (*)	10/100/1000 Mbps (*)	10/100/1000 Mbps (*)
Micro SD Interface	✔	✔	✔	✔
eSata II 3Gbps	✘	✘	✔	✔
RTC	✔	✔	✔	✔
Optical S/PDIF Audio Out	✔	✔	✔	✔
Micro USB to UART	✔	✔	✔	✔
InfraRed for Remote Control	Receiver & Transmitter	Receiver & Transmitter	Receiver & Transmitter	Receiver & Transmitter
Power adapter specification	DC Jack 5.5mm 5V, Max 2A current	DC Jack 5.5mm 5V, Max 2A current	DC Jack 5.5mm 5V, Max 2A current	DC Jack 5.5mm 5V, Max 2A current

Figura 5.6: Comparativa dispositivos CuBoX-i

5.2. Tarjetas de red

Se han evaluado para el presente TFG las diferentes tarjetas de red que hay en el mercado en base a:

- Interfaz de la tarjeta de red:

  Integrada: se integran dentro de la propia placa del dispositivo.

  Interna: se alojan dentro del interior del equipo, pero no son parte del hardware del dispositivo.

  Externa: van conectadas fuera del equipo, a través de un puerto de conexión como un USB.

- Velocidad: las velocidades encontradas para una tarjeta de red en el mercado rondan los 10, 100 y 1000 Mbit/s.

- Estándares WiFi: las tarjetas de red pueden soportar varios estándares. Entre los principales encontrados, están:

  802.11a: en teoría, la velocidad máxima de este estándar es 54Mbit/s. Nosotros nos encontramos con la realidad de 20Mbit/s. La banda es de 2,4Ghz.

  802.11b: en este caso, la velocidad máxima de transmisión de 11Mbit/s. Nosotros nos encontramos con la realidad de unos 5.9Mbit/s-6.1Mbit/s. La banda es de 2,4Ghz.

802.11g: nos ofrece una velocidad máxima teórica de 54Mbit/s al igual que en el estándar 802.11a. Nosotros nos encontramos con la realidad de 21Mbit/s. La banda es de 2,4Ghz.

802.11n: surgida en 2004, cuenta con una velocidad real de hasta los 600 Mbit/s. En este caso, la banda esta tanto para 2,4Ghz como para 5Ghz.

802.11ac: posee una banda de 5Ghz. Las velocidades llegan a alcanzar los 3,46Gbps. No se ha podido probar en este Trabajo de Fin de Grado la velocidad real.

Existen multitud de adaptadores WiFi que pueden ser usados para las sondas. Una opción muy beneficiosa para tener WiFi en los mini ordenadores es hacer uso de mini adaptadores USB WiFi. Esta opción beneficia a la hora de ocupar espacio y no perjudica en cuanto a realizar mayor número de conexiones ocupando el resto de entradas USB de las que dispone el dispositivo.

Tras el estudio realizado podemos afirmar que el chip más potente y recomendado se encuentra en las antenas ALFA AWUS036NHA. Este chipset es el AR9271. Actualmente hay una versión moderna, el AR9271L, de menor coste pero menor potencia.

Se pueden encontrar adaptadores WiFi en cualquier tienda de electrónica. Para el presente trabajo de fin de grado se han usado adaptador WiFi ALFA AWUS036NHA al precio de 30euros y adaptadores WiFi con antena externa por 3,98 euros. Se pueden encontrar adaptadores WiFi mini USB desde un euro en Aliexpress.

Figura 5.7: Adaptador WiFi mini USB

Figura 5.8: Adaptador WiFi ALFA AWUS036NHA

5.3. Baterías externas

Conocidas bajo el nombre de PowerBank, se trata de baterías recargables de material de litio con puertos USB que nos permiten cargar dispositivos electrónicos, como mini ordenadores.

La capacidad de una batería externa se mide en mAh (miliamperio x hora). Esta medida es utilizada para medir la carga eléctrica acumulada en un determinado período de tiempo.

A mayor capacidad de mAh, mayor duración de carga podremos ejercer.

Para el caso de un dispositivo móvil, se requiere de una batería externa de unos 3000 mAh para poder cargarlo al 100 %.

Se pueden adquirir estas baterías en cualquier establecimiento siendo su coste a partir de 1 euro las de menor capacidad de 1000 mAh.

5.4. Sistemas operativos

Desde el principio se tuvo claro la utilización de Software Libre en este trabajo, el cual será liberado con una licencia libre. Es por ello, que el principal sistema operativo que se pretendió usar fue GNU/Linux.

Las opciones que se barajaron y probaron para el caso de la Raspberry Pi son: Raspbian, Arch Linux y Windows IOT Core - este último con la visión de probar más allá y analizar posibles competencias. La decisión final de optar por Raspbian fue por su amplia documentación y soporte.

Para el caso de los CuBoX-i, pese a tener opción de usar Android, como este sistema operativo no entra dentro de nuestro alcance, solo se probaron las distribuciones basadas en Debian (como XBian). Como se comentaba anteriormente, estas se encontraban desactualizadas, lo que nos hizo descartar este tipo de mini ordenadores.

Finalmente, para el caso de los dispositivos GL.iNet, se optó por OpenWRT. Esta decisión fue tomada tras analizar el panorama actual y decompilar y compilar el firmware de los dispositivo WiFi-Pineapple. Como se verá más adelante, se realizó un proceso de ingeniería inversa para entender el funcionamiento de los dispositivos que se encuentran actualmente en el mercado.

5.5. Hardware empleado

Anteriormente vimos el hardware del que disponemos que puede formar parte de WiFi-Pocket. Este hardware puede variar, creando diferentes infraestructuras todas ellas dependientes de distintos factores.

En esta sección nos ponemos en el caso de un prototipo final, con el hardware que hemos implementado para su creación. Este prototipo final, se plantea en un espacio ideal.

5.5.1. Microprocesadores

Para la gestión de la lógica de la aplicación, se ha pensado separar los servicios como el IDS, el servidor web, el servidor DNS, el servidor DHCP y el panel de control, de las sondas. Estos servicios pueden ser ejecutados en un computador común de un usuario, en un servidor o en un computador de pequeño tamaño como los vistos anteriormente.

5.5.1.1. Sondas

Para la creación de las sondas, necesitamos como requisito mínimo que estas puedan disponer de dos interfaces de red. Una de las interfaces de red nos servirá para la recolección de información y la otra para el envío. Si el dispositivo que se

emplee ya cuenta con una interfaz de red nativa, nos estaríamos ahorrando costes y tamaño del dispositivo.

El proyecto WiFi-Pocket busca ser usado por cualquier usuario que necesite emplear esta herramienta. Aunque si bien para la configuración se requieren de unos conocimientos mínimos en informática, el usuario no tiene por qué ser un experto en el campo de la ciberseguridad. Es por ello que, debido a su complejidad de configuración, se descartan los dispositivos GL.iNet. También se descartan los dispositivos CuBoX-i por el software desactualizado y su elevado precio.

Dentro de los dispositivos Raspberry, basandonos en el coste nos encontramos con la Raspberry Pi Zero y la Raspberry Pi 3 Modelo A+. De estas dos, la Raspberry Pi Zero queda descartada al no contar con entradas USB ni módulo WiFi incorporado de forma nativa. Como las sondas no requieren de mayor capacidad de procesado que aquella que las permita esnifar el tráfico y enviar la información, no son necesarios modelos superiores a la Raspberry Pi 3 Modelo A+. Aunque se pueden encontrar de segunda mano modelos inferiores a un precio semejante, nos encontramos que algunos servicios como la VPN, requieren de una mayor complejidad de configuración que modelos superiores.

Concluyendo, el computador ideal para las sondas sería la Raspberry Pi 3 Modelo A+.

5.5.1.2. Otros servicios

Para el caso de los diferentes servicios, estos se pueden externalizar. Por supuesto, podemos volcar todo en local sobre un único dispositivo. Se recomienda emplear el último modelo de la Raspberry Pi con 4GB de memoria RAM.

Para ahorrar costes, se han volcado todos los otros servicios del prototipo de WiFi-Pocket sobre la Raspberry Pi 2 modelo B. Este dispositivo es el más económico que cuenta con una memoria RAM de 1GB y una adecuada velocidad de procesamiento. A su vez, dispone de la misma cantidad de puertos USB que los modelos superiores. Durante las pruebas realizadas, no se han producido cortes en la ejecución de las tareas ni sobrecalentamiento en el dispositivo.

En este aspecto, necesitaremos una interfaz de red para recibir la información de los dispositivos y otra para levantar un punto de acceso y así poder conectarnos al panel de control desde un dispositivo externo como puede ser un teléfono móvil. Luego, dependiendo de los ataques a realizar o la necesidad de las tareas, necesitaremos más de una interfaz de red. En un escenario ideal, con dos interfaces de red más sería necesario. En total se requieren 4 tarjetas de red.

5.5.2. Tarjetas de red

Para las tarjetas de red en el caso del computador que reune los otros servicios, como vimos en el apartado 5.5.1.2, serían necesarias cuatro en total. En el caso del modelo de Raspberry pensada como ideal, la Raspberry Pi 2 modelo B, solo podemos tener las interfaces de red WiFi a través de tarjetas de red externas por USB. Tal y como están posicionadas las entradas USB en la placa, solo podremos conectar dos adaptadores WiFi de tamaño superior al mini USB en horizontal, con dos adaptadores WiFi de tamaño mini USB en vertical.

Para el caso de las sondas, como vimos en el apartado 5.5.1.1, tan solo son necesarias dos interfaces de red. El modelo pensado como ideal, la Raspberry Pi

3 Modelo A+, cuenta con un módulo WiFi implementado de forma nativa. A su vez, cuenta con un puerto USB que nos permite conectar una tarjeta de red. En un aspecto ideal, suponemos un escenario donde el alcance a los diferentes dispositivos no es muy grande y, por tanto, podemos hacer uso de adaptadores WiFi mini USB. Estos nos permiten tener un prototipo compacto.

5.5.3. Baterías externas

En un espacio ideal, el dispositivo podrá ir conectado de forma directa a la corriente eléctrica. Por ello, se descarta el uso de baterias externas. Esto nos permite ahorrar costes y reduce el tamaño físico de los dispositivos. Si el espacio fuese un escenario de Red Team donde se necesitase hacer uso de la información que nos aporta el Blue Team, entonces se necesitaría emplear baterías externas.

Capítulo 6

Metodología

"La tarea del hacker no es destruir, sino utilizar sus conocimientos en favor de la libertad y la igualdad social."
— Johan Manuel Méndez

A continuación se explica detalladamente la metodología que se ha empleado para el desarrollo del presente Trabajo de Fin de Grado y como han sido los pasos que se han seguido.

6.1. Metodología Ágil

Para desarrollar este proyecto, se ha implementado una metodología ágil aplicada al trabajo individual. Este método, comúnmente tiene como objetivo principal mejorar la satisfacción del cliente y la velocidad y calidad del desarrollo. Se han considerado como clientes el desarrollador del presente Trabajo de Fin de Grado y disitntos profesionales que se dedican a la ciberseguridad.

A través de este método, tanto los requerimientos como el alcance se ven modificados durante toda la vida del proyecto. Se ha considerado que era la metodología más apropiada ya que en un principio, se tenía claro que se quería hacer, pero no se sabía hasta dónde se podría llegar o qué pasos habría que seguir para llegar al objetivo buscado. Se ha tratado de realizar un trabajo innovador y de utilidad para la comunidad hacker. Tras una extensa investigación y tras una gran variedad de cambios que se reflejan en el desarrollo de la memoria, se ha conseguido llegar a un proyecto estable del cual se vislumbra mejor su alcance.

6.2. El proceso

Si hay algo que estaba claro, es que aunque hubiese un camino a seguir, se barajarían todos los caminos posibles recopilando feedback, probando el producto, mejorándolo, quitando código, añadiendo más código, ...

WiFi-Pocket ha sido un proyecto en constante desarrollo y cambio, buscando adaptarse lo mejor posible a las necesidades actuales y a la comunidad hacker.

Figura 6.1: Metodología ágil

6.2.1. Comienzo

Para definir los requisitos previos y encaminar un poco el trabajo, se decidió comenzar por una planificación de 3 fases bien diferenciadas. La visión en este punto era tener una pequeña herramienta simple que resultase ser un captive portal visto desde un punto de acceso creado desde la propia herramienta, que guardase los datos que los usuarios escribiesen. Tras el supuesto inicio de sesión, el usuario podría navegar con normalidad. Se podría consultar la información recogida a través del portal web el cual se podría acceder de forma anónima. En este momento se planteó el anonimizar el acceso a través de tor, que no se llegó a realizar por su poca utilidad en el proyecto.

6.2.2. Problemas encontrados en cada vuelta

Uno de los puntos fuertes que tiene la implementación de metodologías ágiles es que estas permiten una rápida respuesta a los cambios a la par que el cliente participa de forma activa en el desarrollo (en este caso, feedback de personas del sector), mientras se van creando versiones estables del "producto".

Inicialmente se llevó a cabo una tarea de investigación que en ningún momento se abandonó durante este Trabajo de Fin de Grado pues en cada avance y prueba, se descubrían nuevos caminos y nuevas soluciones, y se desechaba material ya utilizado o pendiente de implementar.

A continuación, se detallan los cambios que se han ido produciendo a lo largo del recorrido del presente Trabajo de Fin de Grado. Las modificaciones se han agrupado más que por las fases del desarrollo, por los fuertes cambios producidos.

6.2.2.1. Impulso sobre Red Team

Como se ha estado comentando a lo largo de la memoria, una gran parte del tiempo dedicado al presente Trabajo de Fin de Grado ha sido en investigación. Según se iba desarrollando el programa y se iba avanzando, se iban encontrando

herramientas que ya existían. Muchas de estas herramientas no se encuentran realizando una simple búsqueda en Google, sino que requieren de alguien que ya las conozca o de una búsqueda exhaustiva a través de plataformas que alojan código como GitHub y GitLab. Muchos de los descubrimientos han surgido al presentar las versiones estables que se iban creando a personas del área de la ciberseguridad. Gracias a estas personas, se ha podido vislumbrar cuál es el punto fuerte del Trabajo de Fin de Grado, y se ha llegado a una conclusión muy acertada: en temas de Red Team, en cuanto a WiFi se refiere, está todo inventado. Y es una afirmación muy cierta. Por ejemplo, si queremos crear un punto de acceso, ya hay una herramienta denominada hostapd que nos facilita el trabajo. Si queremos realizar un ataque sobre redes Enterprise, ya hay una herramienta denominada hostapd-wpe que nos hace el trabajo. Si queremos realizar ataques mediante ingeniería social, tenemos herramientas como Fluxion o como Social Engineering Tool, que nos hacen el trabajo tan solo dándole a "siguiente" desde una terminal. Si queremos hacer creer a un usuario que está navegando por un sitio legítimo cuando realmente está pasando su información a través de un proxy configurado para ello, ya hay una herramienta que lo hace. Y así con todos los ataques habidos y por haber. Actualmente, como la seguridad de los routers WiFi se ha estancado en cuanto a cifrado se refiere y, como los usuarios siguen teniendo dispositivos o realizando configuraciones inseguras, no se ha innovado en los ataques que se pueden realizar. Aún se está a la espera de que WPA3 sea desplegado.

Ya hay scripts automatizados que se conectan a una red y nos hacen un escaneo de forma automatizada de los dispositivos conectados a ellas. Estos scripts no son nuevos o de este año, sino que llevan años siendo usados por los pentesters y personas dedicadas a la ciberseguridad.

La evolución ha sido exponencial, se han ido buscando los puntos fuertes de los ataques existentes y de las herramientas ya existentes, buscando en todo momento ayudar a la comodidad y servir de utilidad para las personas que, en un futuro, harán uso de la aplicación.

6.2.2.2. Visión final

Como se ha ido explicando a lo largo de esta memoria, esta herramienta no se centra únicamente en el campo del Red Team, sino que este foco se ha ampliado con una parte defensiva. La parte defensiva en cuanto a redes inalámbricas vía WiFi se refiere está poco trabajada. No hay código libre que se mantenga y funcione, que esté disponible. Es por ello, que la actual herramienta cuenta con un apartado para Blue Team. Dentro de este apartado se podrían generar las reglas que se precisen, y observar a través de un panel visual vía web los ataques o posibles ataques que se están recibiendo o se podrían realizar. Este aumento de la visión, del objetivo final, ha supuesto un gran reto, a la par que ha abierto muchas puertas para el desarrollo.

6.2.3. Proyección de futuro

Sobre todo, se ha estructurado el código de forma modular y aplicando diferentes patrones que oermitirán a todo futuro usuario que desee colaborar en el proyecto, adaptarse de forma fácil y añadir mejoras.

Como decimos, WiFi-Pocket espera ser una herramienta de Software Libre usada por la comunidad dedicada a la ciberseguridad. Por esta razón era fundamental

tener un código bien estructurado y pensado para ser realizado implementando diferentes fragmentos que no alterasen la raíz del programa principal.

El uso de JSON a la hora de intercambiar información permite que el lenguaje de programación pueda ser modificado. La implementación del patrón DAO nos permite poder gestionar las BBDD y las conexiones con el programa independientemente de la utilizaciónmde MongoDB o MySQL.

6.3. KanBan

Cuando hablamos de KanBan, hablamos de un método para gestionar el trabajo. Un sistema de gestión visual. Con este método, se representan los elementos de trabajo a modo de tarjetas de trabajo.

Se emplearon tres columnas, a modo de To Do (por hacer), Doing (en desarrollo) y Done (hecho). Esto permite visualizar cómo va avanzando el desarrollo de la aplicación y qué nuevas mejoras o qué nuevos trabajos hay que ir desarrollando. También permite tener un cuaderno de bitácora del trabajo que se va realizando o queda pendiente de realizar.

Para esta gestión se utilizó el apartado proyectos de GitHub, plataforma donde se encuentra el código contenido. Este apartado nos permite tener tareas organizadas en columnas.

Dentro de esta página, se dividió el trabajo en 3 columnas principales. Por un lado, teníamos el proyecto "Memoria", el cual servía para controlar las tareas relacionadas con el desarrollo de la memoria. Por otro lado, teníamos todas las tareas referentes al panel de control de la aplicación. Este panel es el que más tareas ha necesitado para su desarrollo. Finalmente, añadido en las últimas etapas del TFG, se encontraba el proyecto sondas. Este proyecto contenía todas las tareas vinculadas con las sondas.

También, se ha introducido como tareas ideas que convendría implementar de cara a un futuro, en caso de seguir desarrollando el proyecto.

Todas las tareas iban con su etiqueta correspondiente, para identificar mejor el tipo de tarea.

A continuación, en la figura 6.2 se muestra un ejemplo del proyecto "Panel de Control".

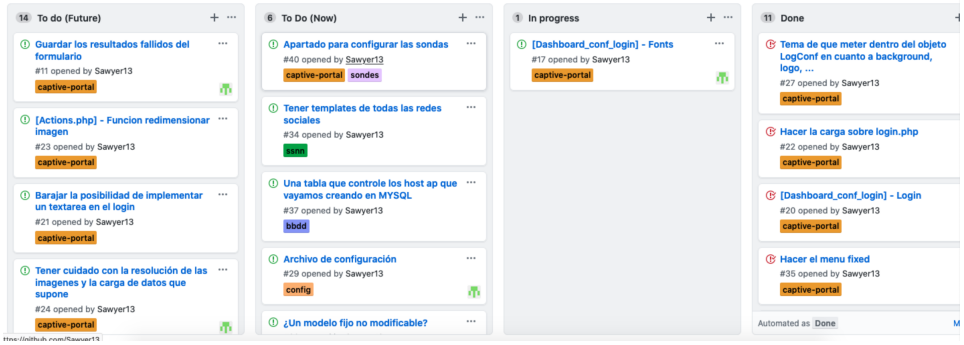


Figura 6.2: Apartado Proyectos de la plataforma GitHub

Capítulo 7

Presentación de resultados

"Mi delito es la curiosidad."
— The Mentor

7.1. Pruebas lógicas

En este apartado se muestran las pruebas que se han realizado haciendo uso del software de WiFi-Pocket.

Al ser pruebas a través de una red inalámbrica como WiFi, se ha procurado tener las mismas características ambientales en todas las pruebas realizadas. Para tener un resultado más exacto, cada prueba ha sido realizada varias veces. Todos los resultados se muestran en las tablas.

Los resultados con respecto a la realización de los ataques a las redes WiFi dependen de varios factores. Algunos de estos factores a tener en cuenta son la complejidad de la contraseña, el número de dispositivos conectados a una red, ..., se ha decidido que no eran representativos y no se han añadido en la presente memoria.

7.1.1. Detección de ataques

Se han realizado varias pruebas sobre las detecciones que nos permite realizar el Sistema de Detección de Intrusos que se ha implementado en WiFi-Pocket.

En estas pruebas, además de detectar la existencia de ataques o de usuarios conectados y de descubrir si los usuarios en la red cuya MAC estuviera en las listas negras, informan de si se ha producido algún error.

Para la realización de estas pruebas, se han emulado varios ataques y varios usuarios. Estos ataquen han sido realizados a la misma distancia de la sonda que captura el tráfico. En total, se ha probado con 10 usuarios con diferente dirección mac de la lista negra, 10 usuarios que enviaban paquetes con el flag activo de Probe Request y 10 puntos de acceso falsos diferentes.

En la figura 7.1 se puede observar el modo debug de la sonda. A través de este modo debug se pueden ir controlando los paquetes que hay al rededor y que son capturados por la sonda.

Figura 7.1: Sonda en modo de debug

La tabla 7.1 muestra si se ha detectado la acción, si la acción ha sido mal detectada o si simplemente no ha sido detectada.

Detección de ataques			
Acción	Detectados	Erróneos	No detectados
Detección de usuarios	10	0	0
Creación de advertencias	10	0	0
Notificación de peligros	10	0	0

Tabla 7.1: Detección de ataques a través del IDS implementado

7.1.2. Recolección de información

Se han realizado varias pruebas sobre la capacidad a la hora de recolección de paquetes que tiene el software implementado en las sondas de WiFi-Pocket. Estas pruebas se han hecho sobre el software desarrollado en Python y sobre el software desarrollado en NodeJS.

Estas pruebas han dado los resultados que muestra la tabla 7.2 (basados en tiempo).

A su vez, se ha realizado una prueba para detectar el número de paquetes que se han perdido o recibido. Esta prueba se ha realizado con 3 tandas de 10 paquetes. Los resultados se muestran en la tabla 7.3

Recolección de información				
	Beacons	Probe Reques	Deauth	Auth
Envío 1 NodeJS	6.307 ms	14.213 ms	6.474 ms	4.487 ms
Envío 2 NodeJS	20.528 ms	4.070 ms	4.036 ms	9.302 ms
Envío 1 Python	41.250 ms	7.110 ms	5.737 ms	5.984 ms
Envío 2 Python	4.982 ms	6.489 ms	5.901 ms ms	7.251 ms

Tabla 7.2: Tiempo de recolección de información a través de las sondas

Recolección de información			
Python		NodeJS	
Recibidos	Perdidos	Recibidos	Perdidos
9	1	10	0
10	0	10	0
10	0	10	0

Tabla 7.3: Estadística de los paquetes recolectados por las sondas

7.2. Divulgación

Como parte del proceso de metodología ágil y, con el objetivo de obtener feedback acerca de la herramienta que se está desarrollando mientras se pone en práctica todo el trabajo realizado, se ha presentado el presente Trabajo de Fin de Grado a través de diversas ponencias, que bien han podido realizarse sobre eventos oficiales, o haber sido realizadas en "petit comité". Se destacan, a continuación, la ponencia realizada para la Semana de la Ciencia organizada por la Comunidad de Madrid y realizada en la Facultad de Informática de la Universidad complutense de Madrid, junto con la ponencia en el evento de ciberseguridad celebrado a nivel nacional CyberCamp, organizado por el Instituto Nacional de Ciberseguridad, el INCIBE.

7.2.1. Ponencia Universidad Complutense de Madrid

- Título: Taller de Hacking - Ciberseguridad
- Descripción: "Se mostrará medidas tanto defensivas como de ataque. Se incluyen conceptos de hacking wifi, cracking de contraseñas, anonimato en la red y protección del usuario y sus comunicaciones."
- Fecha: Lunes 28 de octubre de 2019
- Lugar: Salón de Actos de la Facultad de Informática de la Universidad Complutense, Madrid, España

7.2.2. Ponencia CyberCamp

- Título: Hardware y Software para Redteam y Blueteam en redes WiFi

Figura 7.2: Semana de la Ciencia - Presentación del ponente

- Descripción: "Vivimos en un mundo conectado, un mundo conferido a lo virtual, sin barreras, sin fronteras. Un mundo que nos permite intercambiar información con distintas personas a lo largo del planeta.

  El objetivo de la ponencia se centra en conseguir que los asistentes puedan construirse su propio set multiusos de pentesting a través de herramientas de hardware y de software.

  Por un lado, en la parte defensiva, se busca poder identificar en tiempo real ataques a nuestra infraestructura de red inalámbrica via WiFi. Por otro lado, en la parte ofensiva, podremos identificar mediante pruebas de intrusión las vulnerabilidades de nuestra infraestructura."
- Fecha: Viernes 29 de noviembre de 2019
- Lugar: Palacio de Congresos, Valencia, España

Figura 7.3: Semana de la Ciencia - En directo desde el taller

Figura 7.4: Cybercamp - Preparando la presentación

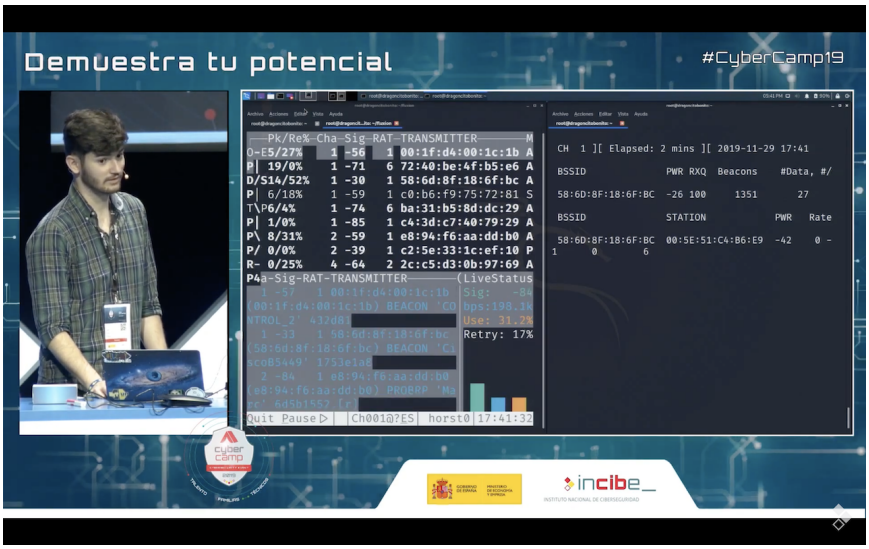


Figura 7.5: Cybercamp - En directo desde el taller

Capítulo 8

Conclusiones y Trabajo Futuro

"Argumentar que no te importa el derecho a la privacidad porque no tienes nada que esconder es como decir que no te importa la liberta de expresión porque no tienes nada que decir."
— Edward Snowden

A continuación, se presentan las conclusiones y el trabajo futuro del presente Trabajo de Fin de Grado.

8.1. Conclusiones

WiFi-Pocket es una herramienta orientada a la ciberseguridad que busca formar parte del arsenal de recursos de Red Team y Blue Team. El objetivo es prevenir posibles fallos de seguridad, a la vez que alerta en tiempo real de ataques que se estén recibiendo sobre una infraestructura.

En la actualidad existe poco desarrollo de medidas de detección de intrusos y generación de alertas de ataques en WiFi. Esto, sumado a la complejidad en el uso de herramientas ofensivas, hace que no todos sean consciente de si están recibiendo un ataque o de si su infraestructura o sus dispositivos son seguros.

Se ha realizado una investigación exhaustiva sobre el protocolo WiFi, observando todos los posibles ataques que pueden realizarse sobre estas redes y sobre los usuarios que navegan por ellas. Además, se han analizado todas las herramientas, tanto de la parte ofensiva como de la parte defensiva, que existen en la actualidad. Para esta investigación, se ha probado distintos dispositivos hardware y una gran variedad de herramientas. Aquellas que no son herramientas de software libre, han sido analizadas a través de un proceso de ingeniería inversa.

Para realizar la labor de Red Team y Blue Team, WiFi-Pocket vuelca su software sobre diferentes dispositivos hardware. Este software está dividido en varias partes. Por un lado, tenemos las sondas, que recogen la información y se la envían al panel central. Por otro lado, tenemos el panel central, que contiene un captive portal que da acceso a una interfaz a través de la cual se pueden gestionar las labores de ataque y de defensa de la herramienta. Todo el tráfico de información va cifrado gracias al uso de una red privada virtual.

Se ha conseguido encajar todas las piezas: bases de datos, servidor web, servidor dns, servidor dhcp, websockets, ... Todo ello empleando dos tipos diferentes de bases de datos y varios lenguajes de programación, cada uno con su propio fin, utilizando patrones de software y formatos específicos para no afectar a la infraestructura haciendo que todo funcione de forma correcta.

En la parte ofensiva, se han implementado 4 tipos diferentes de ataques. A través de WiFi-Pocket podemos desautenticar usuarios, escanear redes, realizar ataques a redes empresariales y crackear contraseñas. Estos ataques pueden combinarse, produciendo mejores resultados. Su interfaz es sencilla y amigable, y puede ser usada por cualquier usuario y desde cualquier dispositivo con un navegador web.

En la parte defensiva, se ha implementado un motor de reglas para la creación de un IDS. Estas reglas detectan, en tiempo real, posibles peligros que pudieran surgir sobre una infraestructura o usuario y posibles ataques que se estén produciendo. También podemos establecer listas negras y blancas de usuarios para detectar posibles accesos no permitidos o controlar accesos. Las alertas pueden verse a través de una interfaz web principal, donde se detectan los ataques en tiempo real.

A su vez, podemos modificar el diseño del captive portal o realizar acciones como la creación de nuevos puntos de acceso. Estos puntos de acceso pueden venir bien en la realización de tareas tanto de ataque como de defensa.

El resultado es una herramienta completa, que posee parte ofensiva y parte defensiva, preparada para ser usada por expertos en seguridad y por no tan expertos, con el objetivo de mejorar la privacidad y seguridad de los usuarios y de las instituciones.

8.2. Trabajo Futuro

Wifi-Pocket puede mejorarse con un mayor número de funcionalidades. Se ha desarrollado esta herramienta de forma modular para permitir al resto de usuarios y/o personas de la comunidad del software Libre añadir código.

Algunas de las posibles futuras implementaciones y mejoras se presentan en los siguientes apartados.

8.2.1. Mejoras en la parte del Red Team

Algunas mejoras que se pueden implementar dentro de la orientación a Red-Team de la herramienta son:

- Aumentar el número de tipos de ataques a realizar.
- Poder realizar ataques a través de las sondas en remoto. Esto permitiría tener sondas destinadas a ataque y a defensa.
- Una opción a barajar es triangular la posición de un atacante basado en la potencia de los puntos de acceso.

8.2.2. Mejoras en la parte del Blue Team

Algunas mejoras que se pueden implementar dentro de la orientación a Blue-Team de la herramienta son:

- Poder escribir las reglas vía panel como si se tratase de Iptables.
- Poder tomar medidas contra el atacante como expulsarle de todos los puntos de acceso a los que intente conectarse o atacar directamente a los nuevos puntos de acceso que pueda haber creado.
- Probar a enviar la información por sockets comprimida para mejorar la velocidad de envío.

8.2.3. Mejoras en la parte general de Wifi-Pocket

Algunas mejoras que se pueden implementar dentro de la orientación a la herramienta en general son:

- Las sondas podrían ampliar el rango de comandos recibidos por socket. Por ejemplo, en caso de tener las sondas conectadas a una batería, pedir por socket el porcentaje de batería que queda.
- Generar reportes en LaTeX que muestren la información tanto de ataques realizados como de ataques recibidos.
- Tener un solo script de instalación. La idea sería que primero se instalan las sondas y luego el panel de control. Esto permitiría, por ejemplo, ir recabando las claves públicas de las sondas para que así, desde el panel de control, se establezca la conexión con todas.
- Habilitar la parte de recogida de datos de las diferentes redes sociales sin dejarlo a dependencia del usuario.

Chapter 8

Conclusions and Future Work

"Arguing that you don't care about privacy because you have nothing to hide is no different than saying you don't care about free speech because you have nothing to say."
— Edward Snowden

Coming up next, the conclusions and future work of this Final Degree Project are going to be presented.

8.1. Conclusions

WiFi-Pocket is a cybersecurity tool that seeks to be part of the resources arsenal of Red Team and Blue Team. The objective is to prevent possible security flaws, while alerting in real time of attacks that are being received on an infrastructure.

Nowadays there is little development of intrusion detection measures and generation of attacks alerts in WiFi. This, coupled with the complexity in the use of offensive tools, makes not everyone aware of whether they are receiving an attack or whether their infrastructure or their devices are safe.

An exhaustive investigation has been carried out on the WiFi protocol, observing all the possible attacks that can be carried out on these networks and on the users that browse through them. In addition, all the tools, both of the offensive and the defensive part, which exist today, have been analyzed. For this investigation, different hardware devices and a variety of tools have been tested. Those that are not free software tools have been analyzed through a reverse engineering process.

To perform the work of Red Team and Blue Team, WiFi-Pocket dumps its software on different hardware devices. This software is divided into several parts. On the one hand, we have the probes, which collect the information and send it to the central panel. On the other hand, we have the central panel, which contains a captive portal that gives access to an interface through which the work of attack and defense of the tool can be managed. All information traffic is encrypted thanks to the use of a virtual private network.

It has managed to fit all the pieces: databases, web server, dns server, dhcp server, websockets, ... All this using two different types of databases and several

programming languages, each with its own purpose, using patterns of specific software and formats so as not to affect the infrastructure by making everything work correctly.

On the offensive side, 4 different types of attacks have been implemented. Through WiFi-Pocket we can unauthenticate users, scan networks, carry out attacks on business networks and crack passwords. These attacks can be combined, producing better results. Its interface is simple and friendly, and can be used by any user and from any device with a web browser.

On the defensive side, a rules engine for the creation of an IDS has been implemented. These rules detect, in real time, possible dangers that could arise on an infrastructure or user and possible attacks that are occurring. We can also establish blacklists and whitelists of users to detect possible unauthorized access or control access. Alerts can be viewed through a main web interface, where attacks are detected in real time.

In turn, we can modify the design of the captive portal or perform actions such as the creation of new access points. These access points can come in handy when performing both attack and defense tasks.

The result is a complete tool, which has an offensive and defensive part, prepared to be used by security experts and not-so-experts, with the aim of improving the privacy and security of users and institutions.

8.2. Future Work

Wifi-Pocket can be improved with a greater number of functionalities. This tool has been developed in a modular way to allow other users and / or people of the free software community to add code.

Some of the possible future implementations and improvements are presented in the following sections.

8.2.1. Improvements in the Red Team part

Some improvements that can be implemented within the RedTeam orientation of the tool are:

- Increase the number of types of attacks to be made
- Be able to carry out attacks through remote probes. This would allow probes for attack and defense.
- One option to shuffle is to triangulate the position of an attacker based on the signal of the access points.

8.2.2. Improvements in the Blue Team part

Some improvements that can be implemented within the BlueTeam orientation of the tool are:

- Be able to write the rules via the control panel.

- Be able to take action against the attacker such as expelling him from all access points to which he attempts to connect or directly attack the new access points he may have created.
- Try sending the information in compressed sockets to improve the sending speed.

8.2.3. Improvements in the general part of WiFi-Pocket

Some improvements that can be implemented within the orientation to the tool in general are:

- The probes could extend their range of commands received by sockets. For example, if you have the probes with a battery, ask sockets for the percentage of battery that remains.
- Generate reports in LaTeX that show information about attacks made and attacks received.
- Have a single installation script. The idea would be that the probes are installed first and then the control panel. This would allow, for example, to collect the public keys (required to configure the vpn) of the probes, permitting to the control panel establish a connection with all the probes in just one click.
- Enable the data collection of the different social by the applications and not by the user.

Bibliografía

El futuro es a lo que cada uno de nosotros va a la velocidad de 60 minutos por hora.

C.S. Lewis

AJAX. Documentación ajax de w3school. 1999-2020. Disponible en `https://www.w3schools.com/js/js_ajax_intro.asp` (último acceso, Enero, 2020).

Baameiro, D. G. Como construir tú propia wifi-pineapple casera. urlhttps://blog.garciabaameiro.com, 2019.

Bootstrap. Documentación bootstrap. ???? Disponible en `https://getbootstrap.com/docs/4.4/getting-started/introduction/` (último acceso, Enero, 2020).

CSS. Documentación css de w3school. 1999-2020. Disponible en `https://www.w3schools.com/css/` (último acceso, Diciembre, 2019).

HTML. Documentación html de w3school. 1999-2020. Disponible en `https://www.w3schools.com/html/` (último acceso, Diciembre, 2019).

JavaScript. Documentación javascript de w3school. 1999-2020. Disponible en `https://www.w3schools.com/js/` (último acceso, Diciembre, 2019).

jQuery. Documentación jquery. ???? Disponible en `https://api.jquery.com` (último acceso, Enero, 2020).

NodeJS. Documentación nodejs. ???? Disponible en `https://nodejs.org/es/docs/` (último acceso, Enero, 2020).

PHP. Documentación php. ???? Disponible en `https://www.php.net/docs.php` (último acceso, Enero, 2020).

Python. Documentación python. 2017. Disponible en `http://docs.python.org.ar/tutorial/3/index.html` (último acceso, Enero, 2020).

Ángel Ramos Varón, A., Muñoz, C. A. B., Hansen, Y. F. y Daswani, D. D. *Hacking práctico de redes Wifi y radiofrecuencia*. Ra-Ma, 2014. ISBN 978-84-9964-296-3. Disponible en `https://ucm.on.worldcat.org/oclc/1026080955` (último acceso, Diciembre, 2019).

Wikipedia (JSON). Entrada: "JSON". Disponible en `http://es.wikipedia.org/wiki/JSON` (último acceso, Diciembre, 2009).

Lo que hacemos en vida, tiene su eco en la eternidad.

Maximus Decimus Meridius