ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
(ICAI)

Doble Máster en Ingeniería de Telecomunicación y Ciberseguridad

HONEYPOTS PARA DISPOSITIVOS IOT

Autor
Andrea Fariña Fernández-Portillo

Dirigido por
Dr. Gregorio López López
Dr. Rafael Palacios Hielscher

Madrid
Julio 2022

Andrea Fariña Fernández-Portillo, declara bajo su responsabilidad, que el Proyecto con título **HONEYPOT PARA DISPOSITIVOS IOT** presentado en la ETS de Ingeniería (ICAI) de la Universidad Pontificia Comillas en el curso académico 2021/22 es de su autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmentey la información que ha sido tomada de otros documentos está debidamente referenciada.

FARIÑA FERNANDEZ PORTILLO ANDREA - 45384429R

Firmado digitalmente por FARIÑA FERNANDEZ PORTILLO ANDREA - 45384429R
Fecha: 2022.06.23 19:38:59 +02'00'

Fdo.: ..... Fecha: 23 / 06 / 2022

Autoriza la entrega:

EL DIRECTOR DEL PROYECTO

Nombre del Director

Fdo.: ..... Fecha: ..... / ..... / .....

V. B. DEL COORDINADOR DE PROYECTOS 23 JUN 2022

Nombre del Coordinador

Fdo.: ..... Fecha: ..... / ..... / .....

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Andrea Fariña Fernández-Portillo **DECLARA** ser el titular de los derechos de propiedad intelectual de la obra: *Honeypot para dispositivos IoT*, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga laLey de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, los derechos de digitalización, de archivo, de reproducción, de distribución y de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

(a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.

(b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.

(c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.

(d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.

(e) Asignar por defecto a estos trabajos una licencia Creative Commons.

(f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

(a) Que la Universidad identifique claramente su nombre como autor de la misma

(b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.

(c) Solicitar la retirada de la obra del repositorio por causa justificada.

(d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

(a) El autor se compromete a:

(b) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.

(c) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.

(d) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

(e) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 23 de junio de 2022

ACEPTA

FARIÑA FERNANDEZ PORTILLO ANDREA - 45384429R

Firmado digitalmente por FARIÑA FERNANDEZ PORTILLO ANDREA - 45384429R
Fecha: 2022.06.23 19:39:22 +02'00'

Fdo.: . . . . . .

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Doble Máster en Ingeniería de Telecomunicación y Ciberseguridad

HONEYPOT PARA DISPOSITIVOS IOT

Autor
Andrea Fariña Fernández-Portillo

Dirigido por
Dr. Gregorio López López
Dr. Rafael Palacios Hielscher

Madrid
Julio 2022

HONEYPOT PARA DISPOSITIVOS IOT

Autor: Fariña Fernández-Portillo, Andrea
Director: Dr. Gregorio López López
Co-Director: Dr. Rafael Palacios Hielscher

Resumen del Proyecto

Implementación de honeypots en un entorno de dispositivos IoT para estudiar los comportamientos y tendencias de los atacantes con el objetivo de poder entender qué puntos críticos existen en este tipo de dispositivos.

Palabras Clave: Honeypot, IoT, Dispositivos wearables, SPA, BLE, DoS, Ciberseguridad

Introducción

Este proyecto se ha realizado en el marco de los objetivos del proyecto RAYUELA de la Unión Europea. Consiste en investigar la seguridad en el ámbito de IoT mediante honeypots. Configurando honeypots se puede conseguir saber qué ataques son más comunes y cómo es el comportamiento de los atacantes para de esta manera saber cómo defenderse de ellos y aumentar así la seguridad en dispositivos IoT. En el marco de este TFM se han analizado los honeypots más utilizados en este ámbito, así como dispositivos IoT de uso común. Se ha diseñado un sistema que implementa todo esto como muestra la Imagen 1.

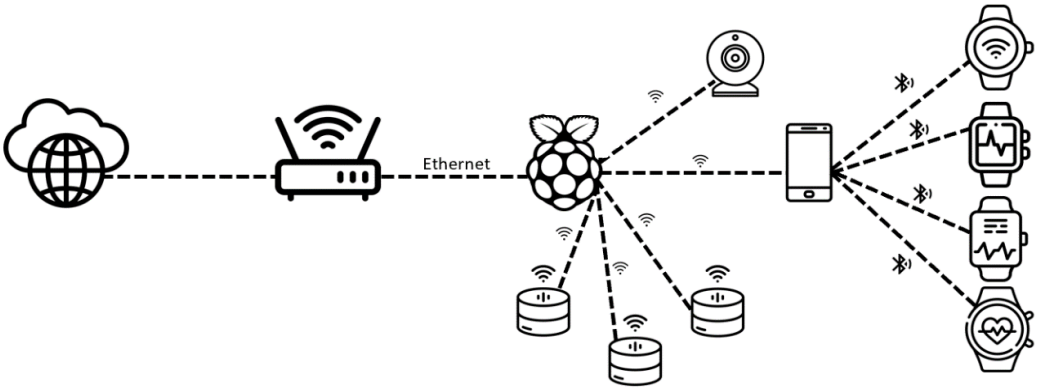


Imagen 1. Esquema del escenario diseñado

Según este esquema, una Raspberry Pi será el punto central de la arquitectura, teniendo conectados a ella múltiples dispositivos IoT, desde una webcam a un smartphone que a su vez conectará con diversos dispositivos wearables, y Smart Personal Assistants.

En la misma Raspberry Pi se implementan los honeypots seleccionados: Cowrie y Dionaea, que registrarán los ataques entrantes desde el exterior. La comunicación con el resto de Internet será posible gracias a un redireccionamiento de puertos en el router.

Definición del proyecto

Este proyecto se ha desarrollado siguiendo tres fases secuenciales.
En una primera fase se analizó el estado del arte del entorno: con qué tipos de dispositivos IoT contábamos, qué es un honeypot, cuáles son sus características, cómo diferenciarlos, … Este estudio ha permitido también conocer qué amenazas suelen sufrir los dispositivos IoT y cuáles son sus vulnerabilidades. Una clasificación de los honeypots según su tipología se muestra en la Imagen 2.

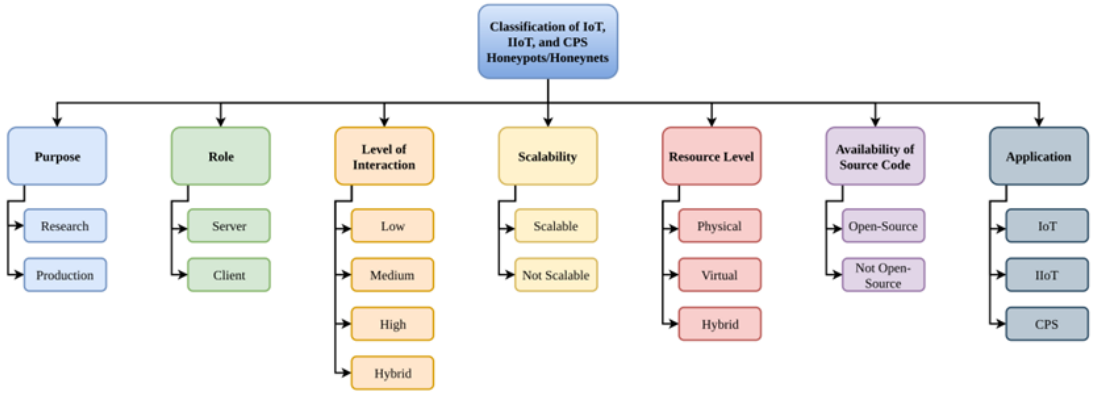


Imagen 2. Tipología de honeypots según sus diferentes características

En una segunda fase del proyecto, se seleccionaron los honeypots a implementar y los dispositivos IoT, así como las conexiones entre los elementos y las configuraciones del sistema para formar una arquitectura completa.

Por último, los resultados registrados por los honeypots se analizaron, además de incorporar Wireshark, un analizador de paquetes para leer el tráfico de los dispositivos IoT.

Resultados

De cada honeypot se extrajo la información relevante al comportamiento que simulaba. De Cowrie se obtienen datos de cómo se ataca el protocolo SSH: de forma distribuida por todo el mundo, sin un país con relevancia destacable y con diversos intentos de login para sortear las acciones de los IDSs.

Dionaea simulaba múltiples protocolos y del análisis de sus registros se obtuvo que el interés principal de los atacantes radica en las bases de datos, siendo MSSQL el servicio más atacado, con un 60% de los ataques registrados.

Utilizado Wireshark, se pudo comprobar que los inicios de comunicación entre los distintos dispositivos IoT con agentes externos, pasan siempre por el servidor correspondiente, por lo que el router aporta una capa de seguridad fundamental, que se rompería si un atacante lograse cambiar la configuración del tráfico del router.

Conclusiones

Las conclusiones a las que se llega gracias a este estudio es que la configuración actual de los dispositivos IoT, que iniciar la comunicación con un dispositivo de fuera de su red a raíz de que el servidor se lo indique, son seguros.

Sin embargo, en líneas de trabajo futuro, se podrían desarrollar los honeypots implementados o utilizar otros adicionales, para simular un sistema más real y de esa forma extraer una mayor cantidad de información de los atacantes.

HONEYPOT PARA DISPOSITIVOS IOT

Author: Fariña Fernández-Portillo, Andrea
Director: Dr. Gregorio López López
Co-Director: Dr. Rafael Palacios Hielscher

Abstract

Implementation of honeypots in an IoT device environment in order to study the behavior and tendency of attackers. This aim is to understand what critical points exist in this type of devices.

Key words: Honeypot, IoT, Wearables, SPA, BLE, DoS, Cibersecurity

Introduction

This project has been carried out within the framework of the objectives of the RAYUELA project of the European Union. It consists of investigating security in the field of IoT through honeypots. By configuring honeypots, it is possible to know which attacks are most common and how the attackers behave in order to know how to defend against them and thus increase the security of IoT devices. Within the framework of this project, the most used honeypots in this field have been analyzed, as well as commonly used IoT devices. A system has been designed that implements all this as shown in Image 1.

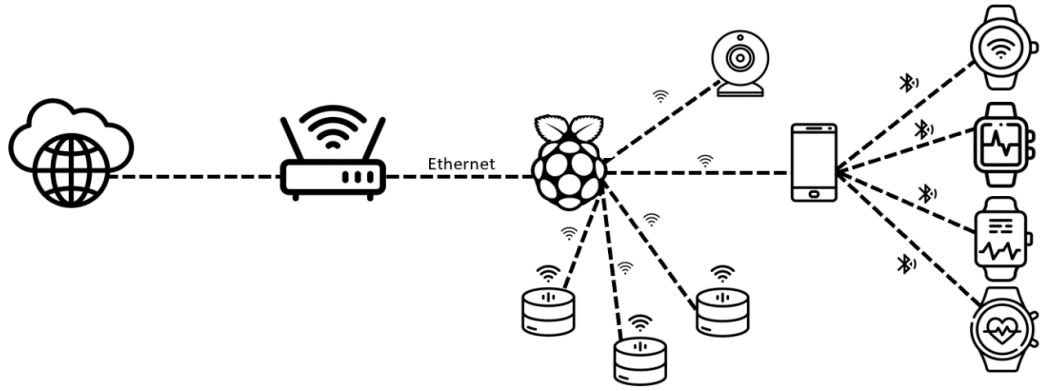


Image 1. Layout of the designed architecture

According to this scheme, a Raspberry Pi will be the central point of the architecture, having multiple IoT devices connected to it, from a webcam to a smartphone, which in turn will connect to various wearable devices, and Smart Personal Assistants.
The selected honeypots are implemented on the Raspberry Pi itself: Cowrie and Dionaea, which will record incoming attacks from outside. Communication with the rest of the Internet will be possible thanks to port forwarding in the router.

Project Definition

This project has been developed following three sequential phases.
In a first phase, state of the art of the environment was analyzed: what types of IoT devices we have, what is a honeypot, what are its characteristics, how to differentiate them, ... This study has also allowed us to know what threats IoT devices usually suffer and which are their vulnerabilities. A classification of the honeypots according to their typology is shown in Image 2.

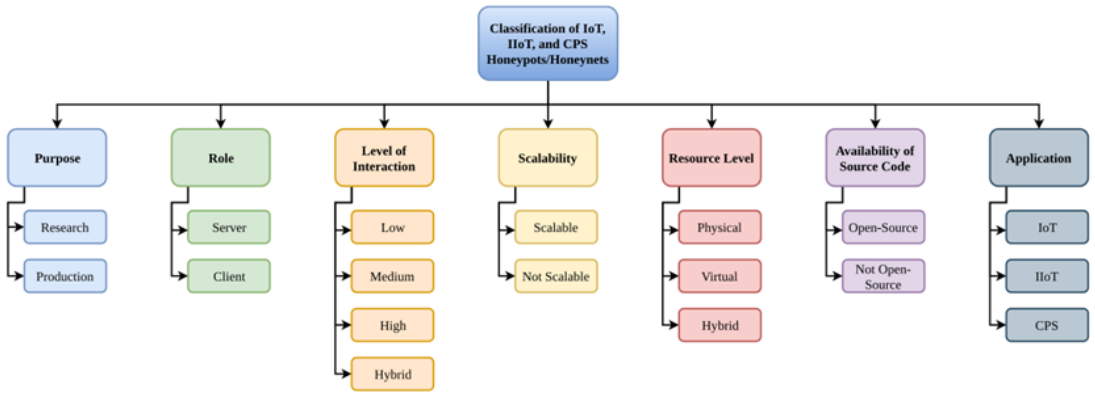


Image 2. Typology of honeypots according to their different characteristics

In a second phase of the project, the honeypots to be implemented and the IoT devices were selected, as well as the connections between the elements and the configurations of the system to form a complete architecture.
Finally, the results recorded by the honeypots were analyzed, as well as the traffic seen by Wireshark, a packet analyzer to read the traffic of the IoT devices.

Results

Information relevant to the behavior of the protocol or service each honeypot simulated was extracted. From Cowrie the data obtained was related to how the SSH protocol is attacked: distributed throughout the world, without a country with notable relevance and with various login attempts to avoid the detection and actions of IDSs.

Dionaea simulated multiple protocols and from the analysis of its logs it was found that the main interest of the attackers lies in databases, with MSSQL being the most attacked service, with 60% of the recorded attacks.

Using Wireshark, it was possible to verify that the beginnings of communication between the different IoT devices with external agents always go through the corresponding server, so the router provides a fundamental security layer, which would be broken if an attacker managed to change the configuration of traffic in the router.

Conclusions

The conclusions reached thanks to this study is that the current configuration of IoT devices, which initiate communication with a device outside their network as a result of the server indicating it, are safe.
However, in future lines of work, the implemented honeypots could be developed or additional ones used to simulate a more real system and thus extract a greater amount of information from the attackers.

A mis padres, por todo.

Quiero agradecer a mis profesores, especialmente a mis directores Rafael Palacios y Gregorio López, por haberme acompañado en este proyecto y a lo largo de mis estudios, confiando siempre en mí.

Por último, quiero dar las gracias a todos mis compañeros y amigos, que me han acompañado en estos últimos años. A Carmen Ollero, por haberme cuidado como una madre, haciéndome ver que mis problemas no eran tan grandes y haber escuchado mis preocupaciones siempre, con cariño y empatía. A Manuel Mora, por haber compartido experiencias y haber estado pendiente y disponible día tras día. A Ignacio Ampuero, por su paciencia y compañía cada día camino a la universidad. Gracias al resto de mis compañeros y amigos.

Gracias a Dios, por la Vida.

Índice general

Índice de figuras

Índice de tablas

Acrónimos

ICAI	Insitituto Católico de Artes e Industrias
TFM	Trabajo Fin de Máster
RAYUELA	empoweRing and educAting YoUng pEople for the internet by PLAying
BLE	Bluetooth Low Energy
DDoS	Distributed Denial of Service
IoT	Internet of Things
SPA	Smart Personal Assistant

Capítulo 1

1. Introducción

En el presente capítulo, se expone el contexto y motivación del proyecto, así como los objetivos y la metodología marcados para su desarrollo y una breve descripción de la organización de esta memoria.

1.1. Motivación

El mercado de dispositivos conectados se encuentra en auge. El número de dispositivos IoT se estimaba en 8.740 millones en 2020, esperando alcanzar los 25.440 millones en 2030 [1], como se muestra en la Figura 1. Sin embargo, aunque este crecimiento acelerado puede favorecer el avance tecnológico en el desarrollo de estos dispositivos, presenta el riesgo de que su producción crezca sin un control ni una regulación adecuados que sean capaces de garantizar unos niveles de privacidad y seguridad mínimos.

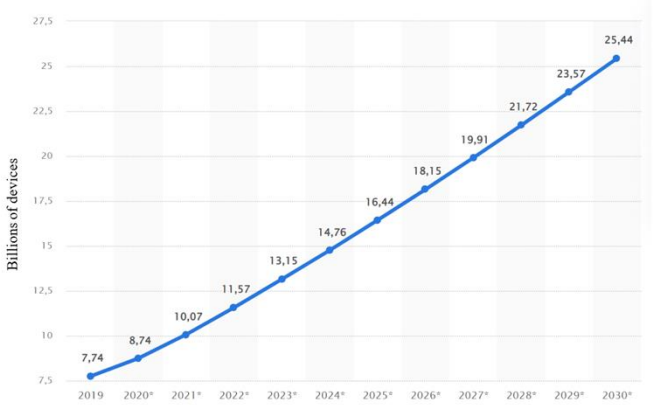


Figura 1. Número global de dispositivos IoT [1]

Esta demanda ascendente hace que la oferta también haya crecido y cada vez son más los diferentes dispositivos IoT disponibles desde asistentes personales inteligentes o SPA (Smart Personal Assistant) hasta cámaras de vigilancia, wearables, etc.

Todos estos dispositivos recogen, generan y almacenan muchos datos que en algunas ocasiones pueden tratarse de información personal sensible de los usuarios y en otras, de la temperatura a la que un usuario quiere tener el termostato de su casa.

Viendo la importancia creciente de estos dispositivos conectados a Internet, surge la motivación de este proyecto: investigar la seguridad en los dispositivos IoT. Para llevar a cabo este estudio, se utilizarán honeypots que permitan recolectar las tendencias, técnicas, tácticas y procedimientos de los atacantes [2].

Para ello, se estudiarán los diferentes dispositivos IoT y honeypots disponibles para construir una arquitectura que permita sacar conclusiones acerca de la seguridad en este entorno. Para que esas conclusiones sean los más realistas posibles, el sistema tendrá que simular de la forma más similar posible, un entorno IoT que disponga de vías de ataque [3].

Con este estudio se obtendrá información para conocer mejor los aspectos más críticos del Internet de las Cosas, qué dispositivos sufren más ataques y por qué. Para poder en un futuro seguir avanzando con en el desarrollo de la seguridad IoT, primero hacen falta estudios como este para conocer cómo funciona.

La motivación de este proyecto surge, por tanto, de la necesidad de comprender esta industria en crecimiento acelerado, donde asegurar la mayor seguridad de los entornos es crucial, mediante un análisis de los dispositivos más vendidos junto a sus aplicaciones, a fin de obtener una perspectiva holística de sus garantías de la seguridad necesaria.

El proyecto desarrollado se alinea con los objetivos del proyecto europeo **RAYUELA** (empoweRing and educAting YoUng pEople for the internet by pLAying) [4], cuyo logotipo se muestra en la Figura 2 y en cuyo marco se ha elaborado un

estudio del estado del arte los honeypots disponibles y de dispositivos IoT, logrando identificar posibles vías de investigaciónque han servido de motivación para este trabajo.

Figura 2. Logotipo del proyecto europeo RAYUELA [5]

1.2. Objetivos

Para poder desarrollar todo lo que se menciona en el apartado anterior, se establecen varios objetivos en este proyecto:

1. Análisis de honeypots. Sabemos qué es un honeypot pero cuántos existen, cómo se implementan, en qué se diferencian, … es información que hay que tener en cuenta antes de decidir cómo se desarrollará el proyecto. Para esto, se recolectará información de otros estudios, se investigarán los honeypots disponibles y sus características, creando una tipología de honeypots que facilite su entendimiento y clasificación.
2. Análisis de dispositivos IoT. Se estudiará cada uno de los dispositivos IoT disponibles y sus características tanto de conectividad como de seguridad.
3. Elección de honeypots y dispositivos IoT. Una vez analizados, se decide qué dispositivos utilizar y lo mismo ocurre con los honeypots. De entre los dispositivos de diferentes categorías disponibles para este proyecto, hay que estudiar cuál sería el objetivo de utilizar cada uno de ellos y si se obtendría información relevante de estos.
4. Diseño de la arquitectura. Con los honeypots y dispositivos que se van a utilizar seleccionados, hay que conformar la visión de toda la arquitectura, cómo será el flujo del tráfico, las conexiones entre dispositivos, etc.

5. Implementación del sistema
6. Recolección y análisis de los resultados obtenidos de los ataques contra los dispositivos.
7. Exposición de las conclusiones del proyecto y proposición de futuras acciones.

1.3. Planificación y Metodología

La Figura 3 presenta un diagrama de Gantt para la planificación y consecución de los objetivos del proyecto.

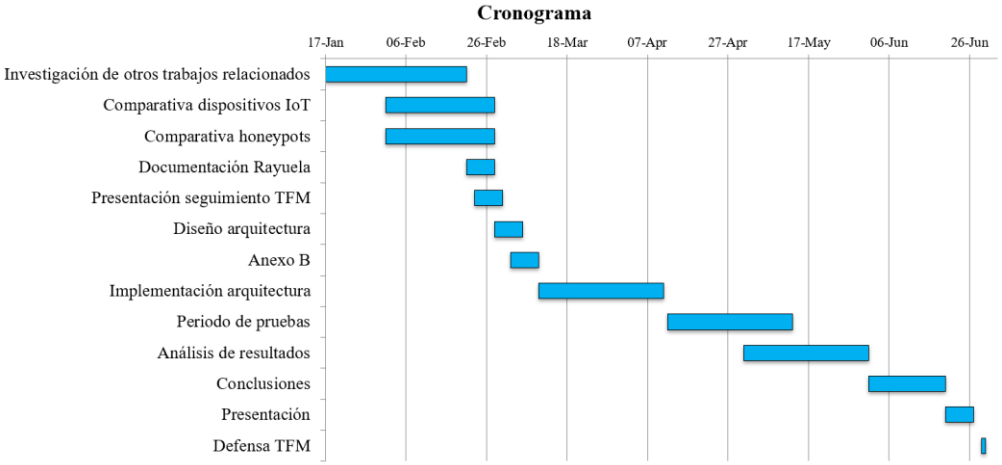


Figura 3. Diagrama de Gantt de la planificación del proyecto

En él se puede apreciar que el trabajo del proyecto se ha dividido en tres fasesdiferenciadas:

- Una primera fase de **investigación**, en la que se ha realizado un estudio del estado del arte de qué es un honeypot y los diferentes tipos que existen, así como de dispositivos IoT.

- Una segunda fase, donde se ha analizado la **arquitectura** idónea para crear un entorno IoT, seleccionando dispositivos y honeypots necesarios.

- Una tercera etapa en la que se ha realizado el **análisis de los resultados** obtenidos y se han extraído **conclusiones** a partir de estos.

Por otro lado, se ha realizado una **documentación** de todo el proceso y resultados del trabajo del día a día, para poder plasmarlo en la memoria presente.

1.4. Organización de la Memoria

En este apartado se va a exponer una guía del contenido de cada capítulo de esta memoria para facilitar su lectura:

- El **capítulo 1** consta de una introducción a la memoria, la motivación de este trabajo, objetivos y metodología seguida a la hora de realizar el proyecto.

- El **capítulo 2** describe el estado del arte de los honeypots, clasificándolos según una tipología y diferentes dispositivos IoT.

- El **capítulo 3** proporciona un análisis del estado actual del mercado de dispositivos wearable, el cual sirve de base para la justificación y presentación de la selección de dispositivos y herramientas del proyecto.

- El **capítulo 4** define el escenario de pruebas establecido para el análisis de los dispositivos seleccionados, y la metodología empleada para el mismo, a la vez que se presentan los resultados de las pruebas llevadas a cabo.

- El **capítulo 5** expone las conclusiones del proyecto y posibles líneas de trabajo futuro.

Capítulo 2

2. Estado del Arte

En el presente capítulo, se describe el estado del arte de las tecnologías y recursos estudiados y empleados a lo largo del proyecto.

2.1. Honeypots

2.1.1. Definición

Un honeypot es un sistema "trampa" o "señuelo", ubicado en una red o en un sistema informático para que su objetivo sea evitar un posible ataque al sistema informático [6]. Aunque tiene muchas funciones, la función principal de un honeypot es detectar y recoger información sobre el ataque informático para así poder detectar de dónde procede y tomar medidas necesarias.

En enero de 1991, mientras Bill Cheswick trabajaba en AT&T Bell Laboratories, descubrió los archivos de registro de los ataques. Tenía la intención de rastrear las pulsaciones de teclas del atacante, aprender sus nuevas técnicas y procedimientos. Después de algún tiempo, Cheswick fue capaz de reconocer al atacante, que se mantuvo en el honeypot hasta que Cheswick lo cerró. El primer honeypot comercial fue lanzado con el nombre CyberCop Sting en 1998 [7].

El término "honeypot" fue iniciado por Lance Spitzner en 1999 en un artículo titulado "To Build A Honeypot" [8], en el que trató de descubrir cómo un atacante realizó un ataque a la red a mediados de la década de 1980. Para ello, construyó a comienzos del año 2000 una red de seis ordenadores en su propia casa. Esta red la diseñó para estudiar el comportamiento y formas de actuación de los atacantes. Fue de los primeros investigadores en adoptar la idea, y hoy es uno de los mayores expertos en honeypots, precursor del proyecto honeynet (www.honeynet.org), en marcha desde 1999 [9].

Su sistema estuvo durante casi un año de prueba, desde abril del 2000 a febrero de 2001, guardando toda la información que se generaba. Los resultados hablaban por sí solos: en los momentos de mayor intensidad de los ataques, comprobaba que las vías de acceso más comunes a los equipos de su casa eran escaneadas desde el exterior de su red, hasta 14 veces al día, utilizando herramientas de ataque automatizadas.

2.1.2. Clasificación

Para poder entender mejor los honeypots, hay diferentes factores a considerar a la hora de clasificarlos y ayudar a seleccionar el más adecuado para implementar dependiendo de su aplicación. Como muestra la Figura 4 [10] estos factores son: objetivo, rol, nivel de interacción, escalabilidad, tipo de recurso, disponibilidad del código fuente y aplicación.

- **Objetivo**: según el propósito que se le dé al honeypot, este puede ser de investigación o de producción. Los honeypots de investigación se utilizan para recolectar y analizar información sobre ataques para desarrollar protecciones contra ellos. Por otro lado, los honeypots de producción se centran más en la defensa, utilizándose para evitar que el atacante acceda al sistema real de la organización que lo implementa.

- **Rol**: según si el honeypot detecta y captura tráfico de forma activa o pasiva. Un cliente honeypot puede iniciar la conversación mediante una petición a un servidor para investigar un programa malicioso, mientras que un honeypot servidor espera a recibir ataques.
- **Nivel de interacción**: según el nivel de interacción que permita el honeypot al atacante, estos pueden clasificarse en honeypots de baja, media o alta interacción. Se trata de la característica más importante. Los honeypots de baja interacción permiten emular servicios con funciones sencillas y no facilitan acceso a un sistema operativo. Sus ventajas son la facilidad de configuración, bajo riesgo, bajo coste y baja necesidad de mantenimiento. Sin embargo, si un honeypot es de baja interacción, el atacante tardará poco en saber que está en un entorno simulado y cancelará su ataque. Por tanto, cuanto más nivel de interacción, más acciones podrá realizar el atacante y, por tanto, se podrá obtener una mayor cantidad de información sobre su ataque. Sin embargo, cuanto más nivel de interacción, más complejo será configurar el honeypot y más coste supondrá. Por eso, se tratará de encontrar un punto medio. Los honeypots de nivel de interacción medio facilitan más servicios que un honeypot de nivel de interacción bajo, incrementando la probabilidad de que no sean detectados fácilmente por los atacantes.
- **Escalabilidad**: hace referencia a la facilidad de un honeypot de crecer e implementar más señuelos. Un honeypot de baja escalabilidad tiene varios señuelos y no es capaz de incrementar sus capacidades. Sin embargo, un honeypot de alta escalabilidad puede extender el número de servicios que ofrece y monitoriza.
- **Tipo de recurso**: los recursos usados para crear un honeypot pueden ser físicos o virtuales. Un honeypot físico está compuesto por aquellos ejecutándose en máquinas físicas, mientras que los virtuales se componen de honeypots virtuales que son hosts de una o más máquinas

físicas. En cuanto a sus características, los físicos son más caros, pero proporcionan mayor interacción al atacante, justo lo contrario que los virtuales. También existen las redes de honeypots híbridas que usan tanto honeypots físicos como virtuales, encontrando un equilibrio entre coste y fiabilidad de las capturas de datos.

- **Disponibilidad del código fuente**: esta característica es importante a la hora de elegir el honeypot a utilizar porque se refiere a la disponibilidad de utilización del honeypot. Cuando el honeypot es open-source, cualquiera tiene acceso a él, puede modificarlo y/o distribuirlo.
- **Aplicación**: según el ámbito en el que el honeypot va a trabajar, este factor se puede clasificar en IoT (Internet of Things), IIoT (Industrial IoT) o CPS (Cyber-Physical Systems).

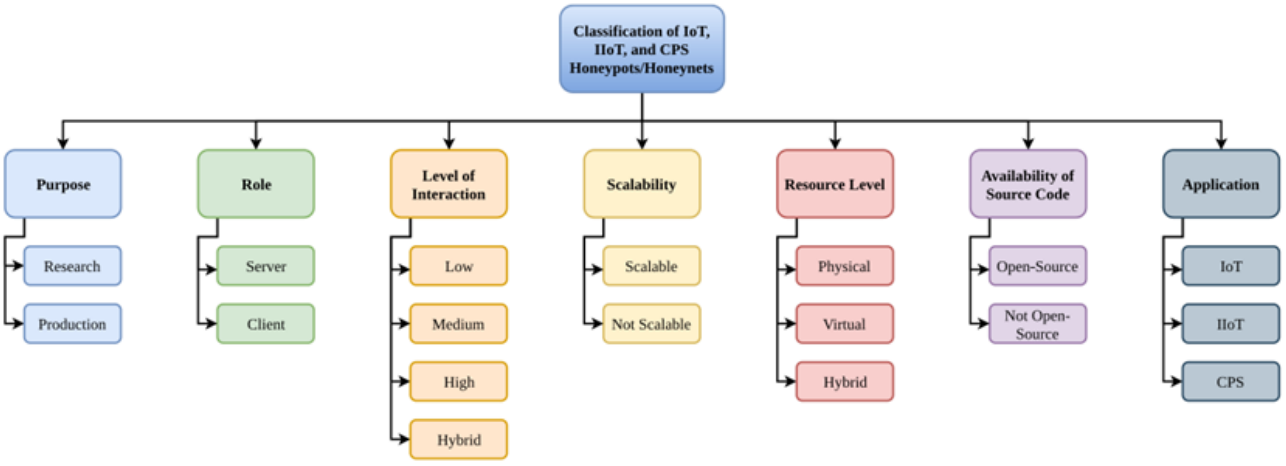


Figura 4. Taxonomía de honeypots [10]

2.1.3. Ejemplos de honeypots para IoT

Según estos factores analizamos los honeypots open-source más populares del mercado en la Tabla I.

Tabla I. ANÁLISIS Y COMPARACIÓN DE LOS HONEYPOTS OPEN-SOURCE MÁS POPULARES

Honeypot	Servicio	Código	Descripción	Ventajas	Desventajas
KIPPO	SSH	Open-source Interacción media	Registra ataques de fuerza bruta e información asociada con la interacción del atacante	Interfaz gráfico útil que muestra la ejecución de comandos exitosa/fallida, URLs visitadas…	No puede simular archivos completos
COWRIE	SSH y Telnet	Open-source Interacción media	Registra ataques de fuerza bruta y captura la interacción realizada por un atacante	- Uso de comandos SCP y SFTP para descargar archivos además de comandos wget o curl - Versión nueva de Kippo - Emulación que permite loggear inicios de sesión de atacantes para un mejor entendimiento de sus TTPs	-
DIONAEA	SIP, FTP, TFTP, SMB, BBDD	Open-source Interacción baja	Emula servicios Intel x86 y ejecución de instrucciones y detecta shellcodes.	Obtiene una copia del malware Tiene soporte para IPv6 y protocolo TLS.	Detección fácil por los atacantes
HONEYD	Crea hosts virtuales en una red	Open-source Interacción baja	Le permite configurar varios hosts virtuales en una red informática con usos de distracción y honeypot	En los registros, puede ver si hay tráfico que se dirige a los hosts virtuales configurados.	Baja interacción
GLASTOPF	Honeypot para aplicaciones web	Open-source Interacción baja	Emula vulnerabilidades de aplicaciones web	Permite recopilar información relacionada con ataques como RFI, LFI, SQLi, …	Baja interacción Foco en aplicaciones web

Aunque no existen honeypots específicamente creados para IoT, los más utilizados en estos entornos por sus aplicaciones, son los mostrados en la tabla anterior:

- **Kippo**: es uno de los honeypots escalables y de interacción media más utilizados. Emula un servicio SSH, que es una de las formas más comúnes de manejar los servicios de manera remota que utilizan los administradores. Kippo detecta intrusiones y ataques de fuerza bruta que se realizan contra la red en la que está implementado [11].

- **Cowrie**: es la versión renovada de Kippo, lo que los programadores

llaman *fork* del código. Las nuevas características que incluye permiten emular los servicios y registrar la sesión de un atacante. De manera que, en base a esos datos registrados de sesiones, se puede sacar conclusiones de las técnicas, tácticas y procedimientos (TTPs) de los atacantes [2].

- Dionaea: nació en 2009 como una versión sucesora de Nepenthes. Se trata de un honeypot de baja interacción que emula servicios de red además de servicios orientados a bases de datos como MySQL, MSSQL. Otros protocolos que simula son HTTP, HTTPS, SMDB, FTP, JETDIRECT. Por todo esto, este honeypot se puede utilizar en el mundo IoT, porque simula conexiones a puertos específicos que serán llamativos para atacantes de dispositivos IoT.

- **HoneyD**: se trata de un pequeño daemon, un programa no interactivo que se encarga de procesos del sistema en un segundo plano [12], que crea hosts virtuales en una red. Se puede considerar un honeypot de baja interacción con buena documentación y se presentó en 2003.

- **Glastopf**: es un honeypot de aplicaciones web de Python. Utiliza la emulación de tipo de vulnerabilidad en lugar de la emulación de vulnerabilidad. La emulación de tipo de ataque popular ya está implementada: inclusión de archivos remotos (RFI) a través de un sandbox de PHP integrado, inclusión de archivos locales (LFI) que proporciona archivos desde un sistema de archivos virtual e inyección de HTML a través de solicitudes POST.

2.2. Dispositivos IoT

2.2.1 Definición

El término *Internet de las Cosas*, o IoT por sus siglas en inglés (Internet of Things), fue usado por primera vez en 1999 cuando Kevin Asthon trabajaba en el campo de la tecnología RFID en red (identificación por radiofrecuencia) y tecnologías de detección emergentes; y explicaba que la sociedad no se basa en ideas o información, sino en cosas [13]. Desde entonces han pasado 23 años y la idea de que el Internet de las Cosas tiene el potencial de cambiar el mundo, como lo ha hecho Internet desde la década de los 80, no ha hecho más que confirmarse.

Es una arquitectura basada en Internet global que facilita el intercambio de bienes y servicios entre redes de la cadena de suministro y que tiene un impacto importante en la seguridad y privacidad de los actores involucrados, por los datos con que trata.

Se prevee que en 2030 haya más de 25 billones de dispositivos conectados a Internet. Estas cifras nos llevan a pensar el porqué del auge de esta tecnología, que radica en la gran inmensidad de posibilidades y aplicaciones que proporciona tanto para la mejora de la vida de los particulares como en las empresas [14].

Existen muchos tipos de dispositivos IoT, al igual que aplicaciones de estos. Para el ámbito de este proyecto se estudian varios.

2.2.2 Smart Personal Assistants

Los Smart Personal Assistants (SPAs) son dispositivos que facilitan el acceso a los usuarios a servicios como e-mail, información del calendario, del tiempo, … usando diálogo con lenguaje natural mediante un dispositivo PDA (Personal Digital Assistant). La interfaz de usuario del SPA debe presentar el

sistema como un único conjunto unificado de asistentes de tareas de back-end, lo que permite al usuario realizar un diálogo en el que es fácil cambiar entre estos dominios [15].

Uno de los aspectos clave de un asistente personal inteligente es su habilidad para organizar y mantener información.

Los ejemplos de asistentes personales inteligentes más populares son Amazon Echo, Google Assistant y Siri de Apple. Cada uno de estos tiene varios dispositivos en los que se ejecuta el asistente. Los que consideramos en este proyecto se muestran en la Tabla II:

Tabla II. DISPOSITIVOS SMART PERSONAL ASSISTANTS

Dispositivo	Tipo	Características	Ventajas	Desventajas
Apple HomePod Mini	SPA	SPA: Siri Se puede configurar si un usuario tiene permiso para agregar o editar dispositivos No se puede interactuar con habilidades de terceros No admite pagos	3º SPA en el mercado	Ecosistema limitado
Google Home Mini	SPA	SPA: Okay Google Los permisos no se pueden establecer por usuario, pero cualquiera puede agregar o editar accesorios Conociendo el proceso de activación, se puede utilizar cualquier acción Los pagos se pueden realizar con un método adicional de autenticación.	Cuota de mercado: 31.4% (2019)[16]	Ecosistema limitado en algunos aspectos
Amazon Echo Show 5	SPA	SPA: Alexa Se ingresan los datos WiFi y se sincronizan los ajustes de Alexa desde el dispositivo móvil El administrador puede habilitar el uso de las aplicaciones de Amazon Alexa, pero cualquier usuario puede activar cualquier habilidad del dispositivo sin confirmación	SPA más utilizado del mercado Permite más acciones que otros SPA	N/A

2.2.3 Dispositivos wearables

Un wearable es un dispositivo electrónico inteligente incorporado a la vestimenta o usado corporalmente como implante o accesorio que puede actuar como extensión del cuerpo o mente del usuario [17].

El origen de los wearables se remonta a hace varios siglos. Desde el inicio

de los tiempos, los humanos hemos intentado mejorar nuestro cuerpo, tanto desde el punto de vista estético como con motivos prácticos.

Una de las principales preocupaciones del ser humano ha sido medir el tiempo. Esto facilita la organización, que nos ha ayudado a civilizarnos y llegar a lo que hoy en día somos. Unos de los dispositivos que intentaron hacer esto de forma portable fueron los huevos de Nuremberg. Se trataba de una especie de huevo, como muestra la Figura 5, que se colocaba a modo de collar, que fue inventado por el relojero Peter Henlein a principios del siglo XVI y se popularizó a partir de 1580 [18]. Se considera que este dispositivo fue el primer "wearable".

Figura 5. Huevo de Nuremberg [18]

A este le siguieron otros dispositivos "vestibles" como el anillo-ábaco en el siglo XVII en China, utilizado para realizar cálculos, tal y como muestra la Figura 6.

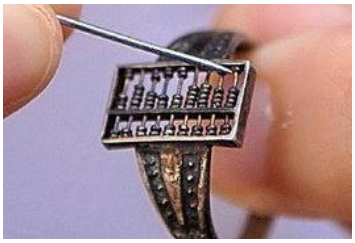


Figura 6. Anillo-ábaco [18]

En 1812 se fabricó el primer reloj de pulsera, que sustituyó al reloj de bolsillo. Y este a su vez comenzó a dar paso al reloj digital a partir de los años 70. Sin embargo, el gran salto de los wearables tal y como los conocemos hoy en día, no ocurrió hasta el siglo XXI, cuando se han empezado a fabricar tecnologías como Bluetooth que han abierto una amplia posibilidad de conexión de estos dispositivos con otros.

Además, en estos años surge una preocupación por la salud y el estado físico de las personas que hasta entonces no había sido tan primordial. De ahí que en 2006 Nike y Apple presentaran en colaboración Nike+, una tecnología que permitía controlar nuestros entrenamientos físicos al milímetro gracias al iPod Nano de Apple y a medidores en las zapatillas Nike.

Dos años después, entra Fitbit en el mercado, una empresa que supone un punto de inflexión en el ámbito de las pulseras inteligentes. Posteriormente, muchas otras empresas surgen con esa tecnología y ofreciendo el producto de Fitbit con diferentes características, proporcionando una gran variedad de opciones y calidades del producto, ofreciendo productos de gama más alta que otros.

El funcionamiento de estas pulseras inteligentes, a diferencia de la mayoría del resto de dispositivos IoT, que implementan comunicaciones Wi-Fi para conectarse directamente a Internet, los wearables generalmente implementan comunicaciones Bluetooth, específicamente Bluetooth Low Energy (BLE). BLE es una tecnología de red de área personal inalámbrica, destinada a aplicaciones en el cuidado de la salud, fitness y beacons, seguridad y las industrias de entretenimiento en el hogar [19]. Comparado con Bluetooth clásico, Bluetooth Low Energy está diseñado para proporcionar un bajo consumo de energía, manteniendo un rango de alcance de comunicación similar [20].

Un dispositivo portátil se conecta a un smartphone a través de BLE. Este teléfono inteligente normalmente ejecuta una aplicación que se conecta a los servidores en la nube de la empresa del wearable o, a veces, de una empresa de terceros, como ilustra la Figura 7. Algunos wearables analizados en este trabajo se muestran en la Tabla III.

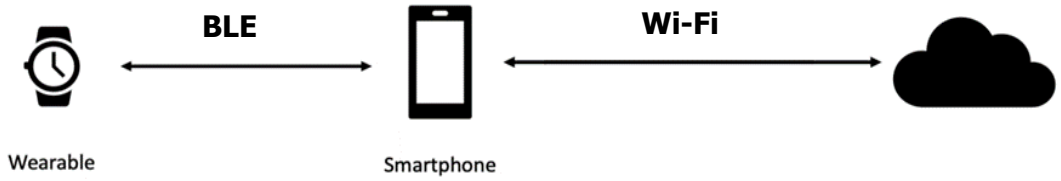


Figura 7. Escenario de comunicaciones de los wearables

Tabla III. DISPOSITIVOS WEARABLES

Dispositivo	App	Descripción
Mi Band 5	Mi Fit	Dispositivo de gama alta Requiere autenticación a través de la aplicación, utilizando una cuenta (Mi cuenta o cuenta de terceros como Google) La aplicación empareja el teléfono con los servidores de Huami y oculta la clave de autenticación en el sistema de archivos del teléfono para que no pueda conectarse con otras aplicaciones. La dirección MAC del dispositivo es estático y se anuncia sin parar cuando no está emparejado.
Garmin vívofit jr.2	Garmin Jr.	Dispositivo de gama alta Requiere autenticación a través de la aplicación, usando una cuenta (puede ser una cuenta de terceros como Google) Falta información sobre la comunicación entre el servidor de aplicaciones. El MAC del dispositivo es estático y se anuncia sin parar cuando no está emparejado.
Fitbit Ace 3	Fitbit	Dispositivo de gama alta La aplicación utiliza Certificate Pinning para evitar el uso de certificados fraudulentos, por lo que no es posible capturar el tráfico HTTP/HTTPS con un MITM
Honor Band 5	Huawei Health	Dispositivo de gama alta La aplicación utiliza Certificate Pinning para evitar el uso de certificados fraudulentos, por lo que no es posible capturar el tráfico HTTP/HTTPS con un MITM
Honor Watch ES	Huawei Health	Igual que la banda de honor 5
BIGGERFIVE Fitness	VeryFitPro	Dispositivo de gama baja No requiere ningún tipo de autenticación o registro
TOOBUR Smartwatch	VeryFitPro	Igual que BIGGERFIVE Fitness

2.2.4 Webcam

Otro de los dispositivos IoT más populares e interesantes en el ámbito de este proyecto de investigación son las webcams, ya que tratan información sensible y muy llamativa para posibles atacantes.

Una cámara de seguridad o doméstica es hoy en día un recurso muy útil para mantener la seguridad en una empresa o una casa, ya que, configurándola con Internet, puede llamar directamente a la policía en caso de que, por ejemplo, se detecte movimiento fuera de un horario establecido.

Por eso este tipo de dispositivos es muy interesante a la hora de investigar su seguridad.

Un ejemplo de webcam de vigilancia de las que tratamos es Dahua Vandal Proof Wi-Fi Dome Camera, que se muestra en la Figura 8:

Figura 8. Dahua Vandal Proof Wi-Fi Dome Camera

2.3. Amenazas y vulnerabilidades en el mundo IoT

2.3.1 Contexto

Como hemos visto, cualquier dispositivo conectado a la red, tiene una dirección IP y, por tanto, puede interactuar con otros dispositivos lejanos y fuera de su red (no contemplamos las IP privadas en este punto). Además, tienen la capacidad de generar, modificar, gestionar y transmitir información.

Por todo eso, todo equipo o dispositivo conectado a Internet y accesible desde cualquier ubicación física, está expuesto a sufrir un ataque o intrusión desde cualquier agente externo. Puede que este intento de interacción sea benigno, con intención de investigar o que por el contrario sea con intención de explotar una vulnerabilidad, robar datos, interceptar comunicaciones u otros muchos objetivos maliciosos.

Por ejemplo, las cámaras de seguridad conectadas a Internet han sido protagonistas recurrentes de brechas de seguridad. Como un arma de doble filo,

un dispositivo que está destinado a la seguridad puede convertirse en lo contrario si no está debidamente cubierto por una capa de ciberseguridad actualizada.

Además, esa información se podría utilizar para saber si hay empleados en las instalaciones o si el lugar está vacío, averiguar la ubicación exacta de una o varias personas, e incluso usar las cámaras para ver las contraseñas y los datos confidenciales, convirtiéndose en un riesgo en el que se puede ver comprometida la seguridad de las personas.

Para entender el estado del arte de la seguridad en estos entornos, hay que entender cómo son atacados estos entornos y qué vulnerabilidades tienen.

2.3.2 Principales ataques

- **Ataques DDoS (Distributed Denial of Service)**: se considera uno de los ataques más peligrosos. Tiene lugar cuando una red de muchos dispositivos conectados a Internet, botnets, generan tal cantidad de peticiones TCP/UDP que saturan el tráfico y colapsan los recursos a los que hacen peticiones, que pueden ser de empresa o DNS públicos. Un caso de estos ocurrió con la botnet Mirai [21] que consiguió tirar más de 50 webs de sitios como Twitter, Amazon…en 2016.
- **Ataques de fuerza bruta** [22]: sin duda es también uno de los ataques más extendidos en dispositivos IoT. Con intención de acceder a los dispositivos IoT, los atacantes utilizan la fuerza bruta, es decir, la repetición constante de intentos de acceso, probando contraseñas genéricas, más comunes y aleatorias.
- **Robo de información**: si los atacantes consiguen interceptan las comunicaciones con información entre dispositivos, podrían obtener datos del uso que se le da a esos dispositivos, historial de navegación…
- **Malware**: el envío de malware a nuestro dispositivo IoT abriría una gran

gama de posibilidades al atacante, desde cambiar la configuración del dispositivo, poder activar la cámara sin que el usuario lo supiera, … Un ejemplo de malware es el ransomware, que consiste en cifrar los datos del dispositivo hasta que la víctima pague un rescate. EL malware podría entrar o llegar a la red a través de Internet e infectar otros dispositivos conectados.

2.3.3 Principales vulnerabilidades

Todas las amenazas mencionadas previamente no supondrían un peligro si las vulnerabilidades por las que conseguir atacar a los dispositivos no existieran. Sin embargo, nunca estamos libres de vulnerabilidades, el riesgo nunca es nulo. Las principales vulnerabilidades de estos dispositivos, aunque puedan parecer sencillas de solucionar, son las que, por error humano, suelen causar más daño y facilitar que las explotaciones de las amenazas tengan lugar de forma exitosa.

- **Contraseñas débiles o por defecto**: muchos de los dispositivos IoT incluyen en su documentación, información de inicio de sesión para configurarlo. Estas credenciales, dadas y públicas, pues suelen ser fácilmente localizables en Internet, están pensadas para que se modifiquen una vez se empiece a utilizar el dispositivo. Los usuarios y contraseñas por defecto son algo que los fabricantes utilizan para poder distribuir masivamente y de forma genérica sus dispositivos. Las credenciales suelen ser algo como usuario: *admin*, *user*, *root* y contraseña: *admin*, *admin1234*, *password*… Esto no debería suponer un problema si los usuarios cambiaran las credenciales, pero pocas veces lo hacen correctamente.
- **Comunicaciones sin cifrado**: si las comunicaciones de estos dispositivos no van cifradas, todo el tráfico viaja en claro y si un agente

intermedio ("Man-in-the-Middle") las interceptara, podría conocer su contenido directamente. Esto supone un riesgo a la hora de que los atacantes puedan generar paquetes fraudulentos, modificarlos…

- **Software desactualizado**: muchos de estos dispositivos cuentan con software en los que en ocasiones se descubren vulnerabilidades. Por lo general, estas vulnerabilidades son solventadas rápidamente en actualizaciones del mismo. Sin embargo, si los dispositivos no actualizan el software, seguirá ocurriendo el mismo problema. Por eso, la falta de actualizaciones del SW supone una de las vulnerabilidades más importantes de los dispositivos IoT.

Capítulo 3

3. Diseño e Implementación de la Arquitectura

En el presente capítulo, se describe el diseño de la arquitectura seleccionada, los dispositivos que la forman y los honeypots desplegados en esta. Además, se expondrá el proceso de implementación de todo el sistema y cómo son las conexiones entre las diferentes partes.

3.1. Diseño

Para conseguir un escenario que incluya todo lo mencionado anteriormente, se propone utilizar una Raspberry Pi 3, que será el punto central de la arquitectura.

La mayoría de los dispositivos irán conectados a la Raspberry Pi por Wi-Fi. Los wearables, como se mencionaba anteriormente, van conectados a un smartphone, pero este a su vez irá conectado por Wi-Fi a la Raspberry Pi. Para ello, hay que configurar un punto de acceso Wi-Fi en la Rasberry Pi.

Para que el sistema tenga acceso y sea accesible desde el exterior, la Raspberry Pi irá conectada al router local mediante conexión Ethernet.

Los honeypots seleccionados se implementarán en la Raspberry Pi, a la

entrada del acceso a los dispositivos IoT. Los dos honeypots que se van a implementar en este proyecto son cowrie y dionaea, debido a los servicios que ofrecen. Cowrie nos facilitará un servicio de SSH y Dionaea otros como HTTP, MYSQL, FTP…

Como se muestra en la Figura 9, la Raspberry Pi utilizará dos tarjetas: wlan0, que es la tarjeta Wi-Fi inalámbrica, al que se conectan todos los dispositivos IoT, y una conexión por cable, eth0:

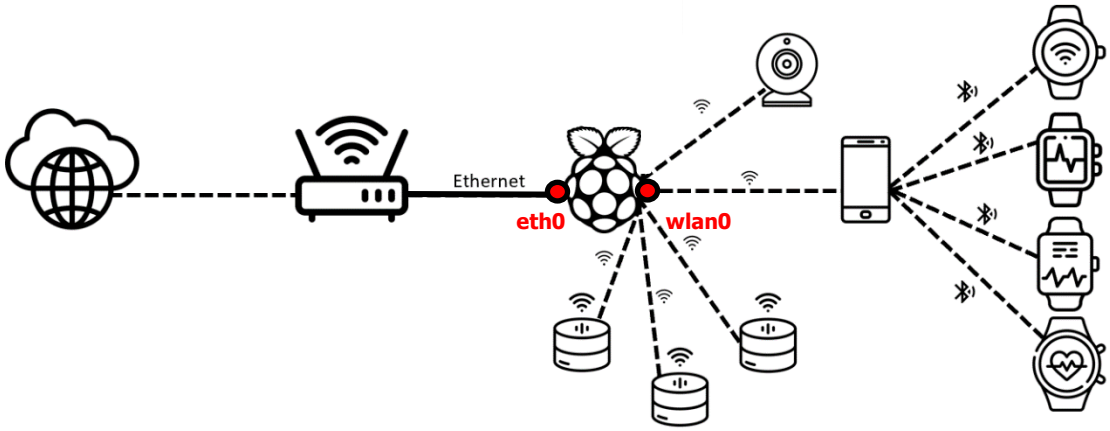


Figura 9. Arquitectura del sistema

3.2. Implementación

3.2.1 Raspberry Pi

Será el elemento central de la arquitectura del sistema. Por un lado, se configura físicamente, conectándolo a la toma de corriente y con un cable Ethernet al router local, para que pueda tener conexión a Internet y se pueda configurar el punto de acceso Wi-Fi para que se conecten los dispositivos IoT. Por otro lado, se implementarán los dos honeypots para emular servicios y registrar las interacciones desde el exterior.

Para configurar el punto de acceso WiFi en la Raspberry Pi, una vez esta tiene conexión a Internet, se hace uso de dos paquetes: hostapd y dnsmasq.

Hostapd es el paquete que permite utilizar un dispositivo WiFi como un punto de acceso, en nuestro caso, se utiliza para convertir el WiFi de nuestra Raspberry Pi en un punto de acceso.

Dnsmasq actúa como servidor DNS y DHCP para que se puedan asignar direcciones IP y procesar peticiones DNS a través de la propia Raspberry Pi.

- Para la configuración del punto de acceso, en primer lugar, actualizaremos la Raspberry Pi:

```
sudo apt update

sudo apt upgrade
```

- A continuación, instalamos los paquetes. Con el siguiente comando instalaremos hostapd, dnsmasq e iptables:

```
sudo apt install hostapd dnsmasq iptables
```

- Paramos los servicios hostapd y dnsmasq para cambiar su configuración y tomar control del interfaz wlan0:

```
sudo systemctl stop hostapd

sudo systemctl stop dnsmasq
```

- Editamos el fichero de configuración DHCP (`sudo nano /etc/dhcpcd.conf`) y añadimos las siguientes líneas y a continuación reiniciamos el servicio DHCP (`sudo systemctl restart dhcpcd`):

```
interface wlan0

    static ip_address=192.168.220.1/24

    nohook wpa_supplicant
```

- Configuramos ahora hostapd, para indicar cómo se interactúa con el dispositivo wlan y determinar el nombre SSID del punto de acceso (*PI3-AP*) y su contraseña (*pimylifeup*):

```
sudo nano /etc/hostapd/hostapd.conf
```

```
interface=wlan0

driver=nl80211


hw_mode=g

channel=6

ieee80211n=1

wmm_enabled=0

macaddr_acl=0

ignore_broadcast_ssid=0


auth_algs=1

wpa=2

wpa_key_mgmt=WPA-PSK

wpa_pairwise=TKIP

rsn_pairwise=CCMP


# This is the name of the network

ssid=Pi3-AP

# The network passphrase

wpa_passphrase=pimylifeup
```

- Configuramos otros dos ficheros, */etc/default/hostapd* y */etc/init.d/hostapd*, añadiendo la siguiente línea:

```
DAEMON_CONF=/etc/hostapd/hostapd.conf
```

- Ahora configuramos dnsmasq. Para ello, movemos el fichero de configuración y lo renombramos:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

- Editamos el fichero de configuración (`sudo nano /etc/dnsmasq.conf`) añadiendo las siguientes líneas:

```
interface=wlan0       # Use interface wlan0
server=1.1.1.1       # Use Cloudflare DNS
dhcp-range=192.168.220.50,192.168.220.150,12h # IP range and lease time
```

- Configuramos el forward del tráfico editando el fichero */etc/sysctl.conf* y añadiendo la siguiente línea:

```
net.ipv4.ip_forward=1
```

- Para activar: `sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"`

- Configuramos la conexión a Internet, haciendo que todo el tráfico de nuestro punto de acceso vaya a la conexión Ethernet:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

- Guardamos las reglas para que no se borren cada vez que se reinicie la Rasperry Pi:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

- Editamos el fichero /etc/rc.local (`sudo nano /etc/rc.local`) y añadimos la siguiente línea antes de la línea `exit 0`:

```
iptables-restore < /etc/iptables.ipv4.nat
```

- Finalmente, se inician los dos servicios y se habilitan en systemctl:

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl start hostapd
sudo service dnsmasq start
```

Comprobamos que se detecta ya la red Wi-Fi con SSID: PI3-AP, e introduciendo la clave previamente establecida, conectamos los diferentes dispositivos IoT.

A continuación, pasamos a la implementación de los honeypots en la Raspberry Pi. Estos van a simular servicios, y actuarán como cebo para los atacantes. Estos honeypots se implementan en la entrada a la Raspberry Pi, es decir, en el interfaz de red eth0, con IP: 192.168.1.48.

Por un lado, se instala Dionaea [23], que emulará varios servicios. Dionaea estará instalado en el directorio */opt/dionaea*. En este directorio encontramos el binario de ejecución */bin*, los logs se almacenan en el directorio */var* y en */etc/dionaea/* está el fichero de configuración *dionaea.cfg*. En la Figura 10 se muestran los servicios levantados antes de la ejecución de Dionaea.

```
pi@raspberrypi:/opt/dionaea/etc/dionaea $ nmap 192.168.1.48
Starting Nmap 7.70 ( https://nmap.org ) at 2022-04-05 12:44 CEST
Nmap scan report for 192.168.1.48
Host is up (0.00088s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
3000/tcp open  ppp

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

Figura 10. Procesos activos antes de la implementación de los honeypots

Para ejecutar el honeypot, primero se comprueba que no haya ninguna instancia ya levantada, con los comandos *ps* y *grep*, tal y como muestra la Figura 11.

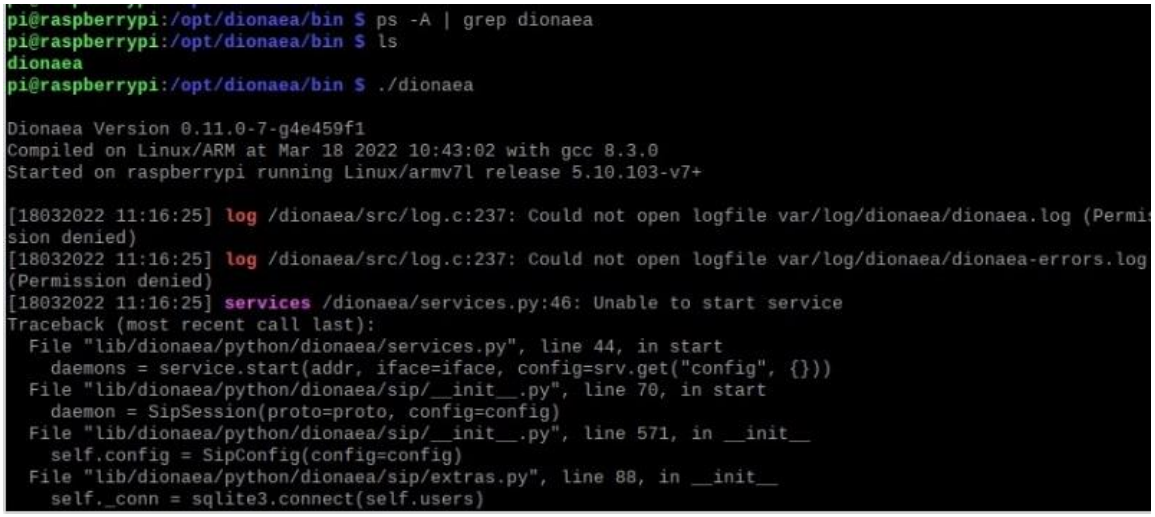


Figura 11. Arranque de Dionaea

En la Tabla IV, se muestran los servicios activos tras la ejecución de Dionaea.

Tabla IV. SERVICIOS Y PUERTOS CORRESPONDIENTES TRAS LA IMPLEMENTACIÓN DE DIONAEA

SERVICIO	PUERTO
FTP	21
NAMESERVER	42
HTTP	80
MSRPC	135
HTTPS	443
MICROSOFT-DS	445
MS-SQL-S	1433
PPTP	1723
MYSQL	3306
SIP	5060
SIP-TLS	5061
JETDIRECT	9100

Por último, implementamos el honeypot Cowrie [24], descargado de http://github.com/micheloosterhof/cowrie, que facilitará un servicio SSH, para estudiar cómo se comportan los atacantes con respecto a este servicio. Para esto, se instalan las dependencias necesarias y se crea un usuario *cowrie* encargado de ejecutar este honeypot. Cowrie se encuentra en el directorio */home/cowrie/cowrie*.

En la Figura 12 se comprueba el acceso por SSH a la Raspberry Pi con el usuario que acaba de ser creado, a la IP donde se encuentra el honeypot, desde otro dispositivo que se encuentra en la misma red privada, porque con la configuración actual, no es posible acceder a los servicios desde Internet. Para hacer esto posible, se llevarán a cabo unos cambios en la configuración del router más adelante.

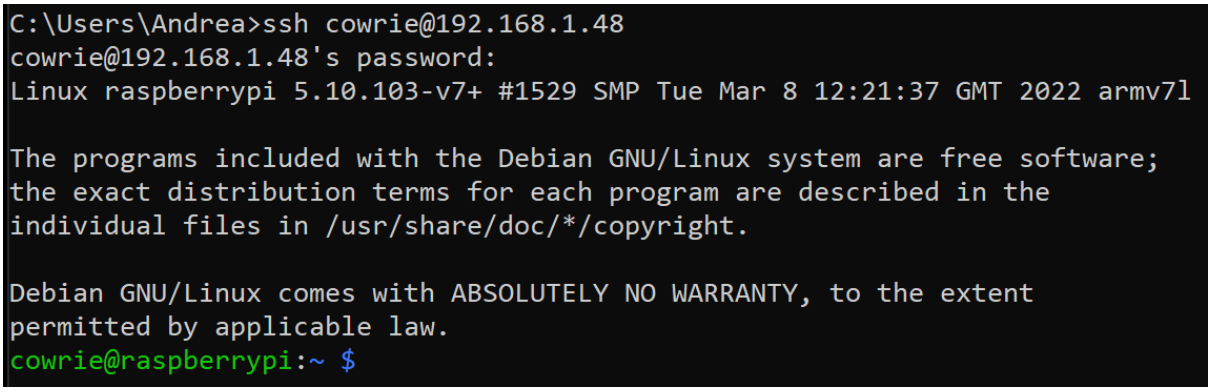


Figura 12. Acceso al servicio SSH de Cowrie desde otro dispositivo en la misma red

3.2.2 Dispositivos IoT

Una vez tenemos configurado el punto central de la arquitectura, pasamos a conectar los dispositivos IoT.

- **Wearables**

Por un lado, conectamos el móvil Huawei por Wi-Fi al punto de acceso PI3-AP, al que se conectan los wearables mediante BLE y que se gestionan desde las distintas aplicaciones. Para ello, descargamos en el smartphone las siguientes aplicaciones: Fitbit, VeryFitPro, Huawei Health…cada aplicación correspondiente al wearable a conectar.

A modo de ejemplo, para conectar Fitbit Ace 3 al smartphone, accedemos a la aplicación de Fitbit y el smartphone hace una búsqueda de dispositivos por BLE y detecta el wearable. En la pantalla del wearable se genera un código que hay que introducir en la app móvil para terminar de sincronizar el wearable con el smartphone. A partir de ahí, ambos dispositivos están conectados. Toda la información recibida por el smartphone se conectará con los servidores de Fitbit por Wi-Fi, a través del punto de acceso Wi-Fi de la Raspberry Pi.

- **Webcam**

El siguiente dispositivo a conectar a la arquitectura es la webcam Dahua Vandal Proof Wi-Fi Dome Camera. Para ello, seguimos los siguientes pasos:

1. Encendemos el dispositivo, conectándolo con el adaptador de potencia.
2. Conectamos por cable Ethernet la webcam al router.
3. Para poder acceder a su configuración, tenemos que encontrar su IP. Para esto, se comprueba la configuración de IPs de la red, que como hemos comprobado anteriormente, es la 192.168.1.0/24. Haciendo una búsqueda de los dispositivos en esa red, descubrimos la webcam, como se muestra en Figura 13.

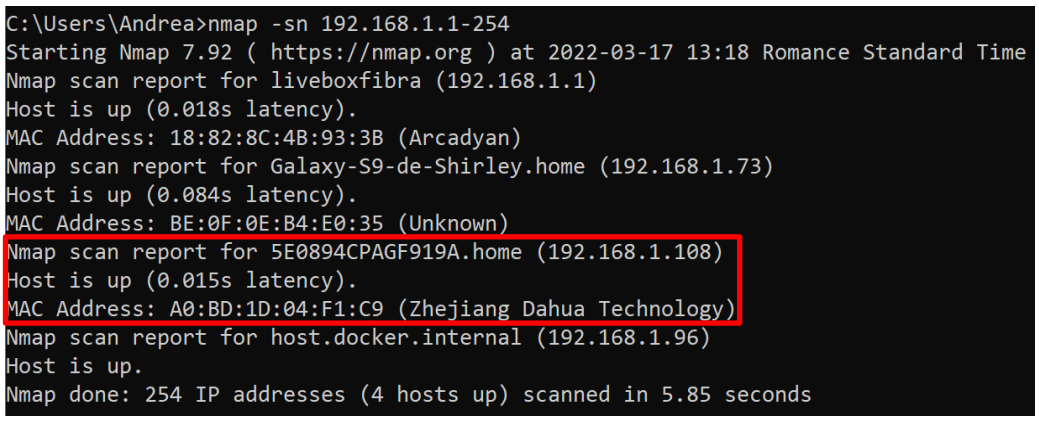


Figura 13. Descubrimiento de la webcam en la red

4. Accediendo desde el navegador a la IP de la webcam, accedemos a su interfaz de configuración, que requiere de una contraseña.

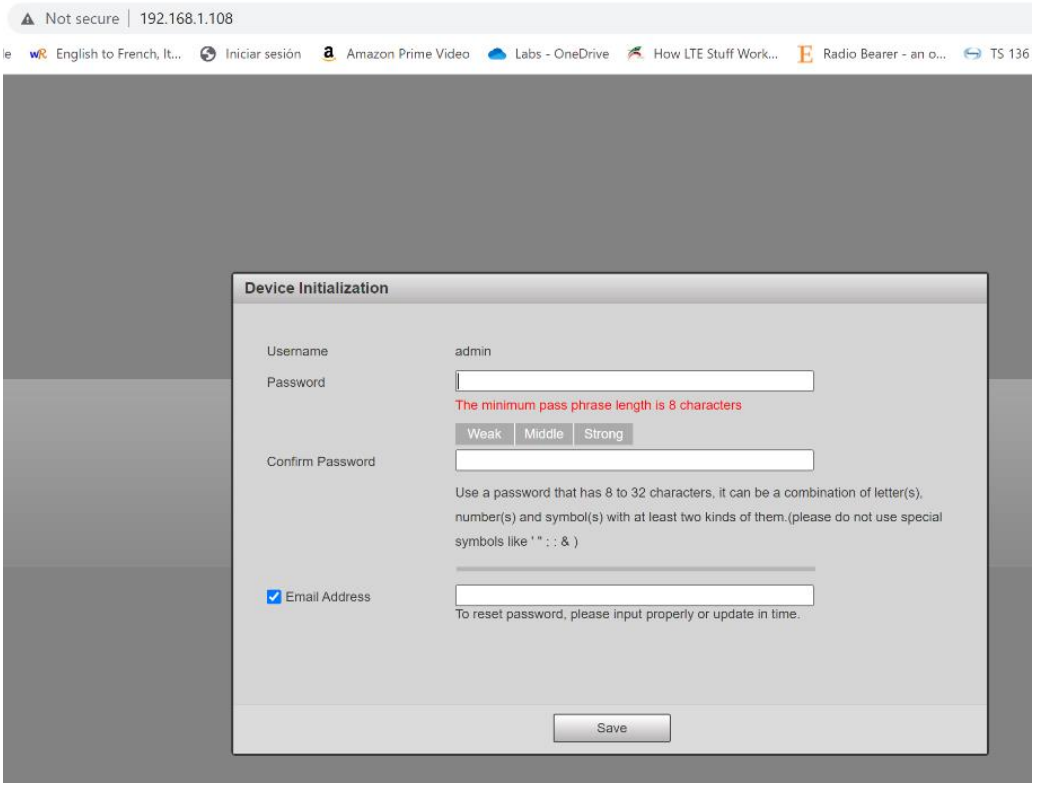


Figura 14. Interfaz de acceso a la configuración de la webcam

Como mencionábamos en el capítulo 2, este es uno de los casos en los que se requiere una contraseña para acceder a la configuración de un dispositivo IoT, y por defecto, esta credencial no está definida al inicializar el dispositivo.

5. Desde el interfaz de configuración, ya se puede conectar el dispositivo a la red Wi-Fi que queramos. En la Figura 15 se muestra el interfaz de configuración cuando la webcam está conectada al Wi-Fi local *MiFibra-933A*, antes de conectarlo al punto de acceso de la Raspberry Pi, su IP era 192.168.1.12. Cuando se conecta a PI3-AP, su IP pasa a ser 192.168.220.142.

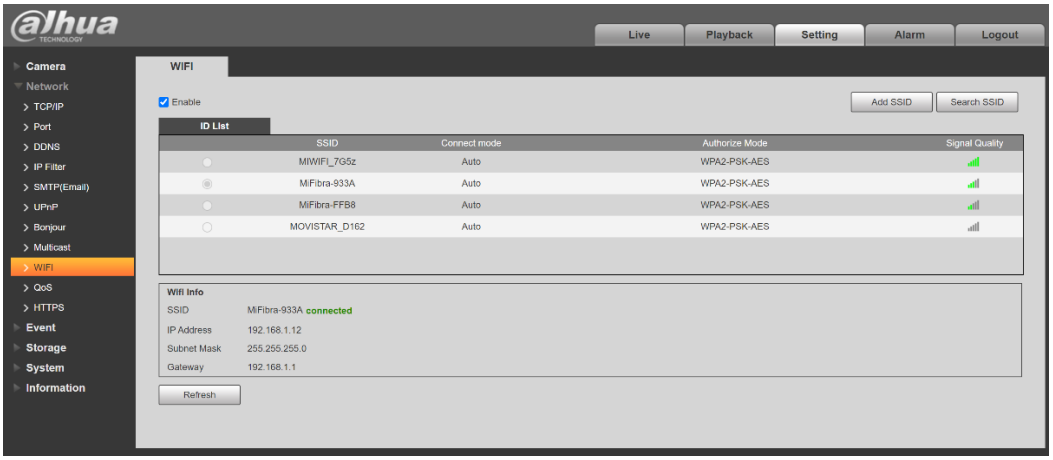


Figura 15. Interfaz de configuración Wi-Fi de la webcam

6. Una vez conectada al Wi-Fi, si desconectamos el cable Ethernet de la webcam, seguirá funcionando.
7. Desde cualquier smartphone con acceso a las tiendas de aplicaciones, podemos descargar la app móvil *DMSS* y añadir el dispositivo para ver el video de forma remota, aunque antes habrá que hacer unos cambios en la configuración del router como se mencionaba en el caso de los wearables.

   El funcionamiento del video se muestra en la Figura 16.

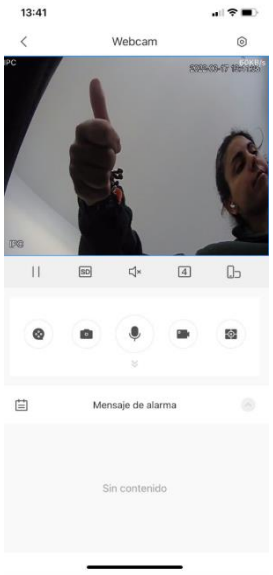


Figura 16. Funcionamiento de la webcam

- **Smart Personal Assistants**

Antes de configurar los diferentes SPAs, creo una cuenta de Gmail para gestionar todos estos dispositivos. El nombre de la cuenta es *tfm.honeypot@gmail.com*.

A modo de ejemplo, se expone paso a paso la configuración del Amazon Echo Show 5:

1. Conectamos el dispositivo a la corriente.
2. Realizamos la selección de la configuración del idioma.
3. En la búsqueda de redes inalámbricas seleccionamos *PI3-AP.*
4. Introducimos la cuenta de Gmail que hemos creado con este propósito.
5. Se añade ubicación, que necesita el SPA para poder dar servicios como la predicción del tiempo. Se indica la sala donde está el SPA y el nombre del dispositivo.
6. Se aceptan los demás permisos y política de privacidad para finalizar el proceso de configuración.

Para configurar el Google Home Mini, se descarga la app móvil Google Home y desde ahí se siguen los pasos de configuración, hasta conectarla al Wi-Fi *PI3-AP*.

En el caso del Apple HomePod Mini, un IPhone lo detecta instantáneamente, pero a la hora de configurarlo daba error de configuración y no ser pudo llevar a cabo.

Por tanto, finalizado el proceso de configuración de todos los dispositivos IoT, se comprueban los dispositivos en la red en la que está el Wi-Fi de la Raspberry Pi, como muestra la Figura 17. Los dispositivos IoT conectados tienen las siguientes IPs:

- **Amazon Echo Show 5** (SPA): 192.168.220.119
- **Huawei P40** (smartphone al que se conectan los wearables por BLE): 192.168.220.131

- **Webcam**: 192.168.220.142
- **Google Home Mini (SPA)**: 192.168.220.143

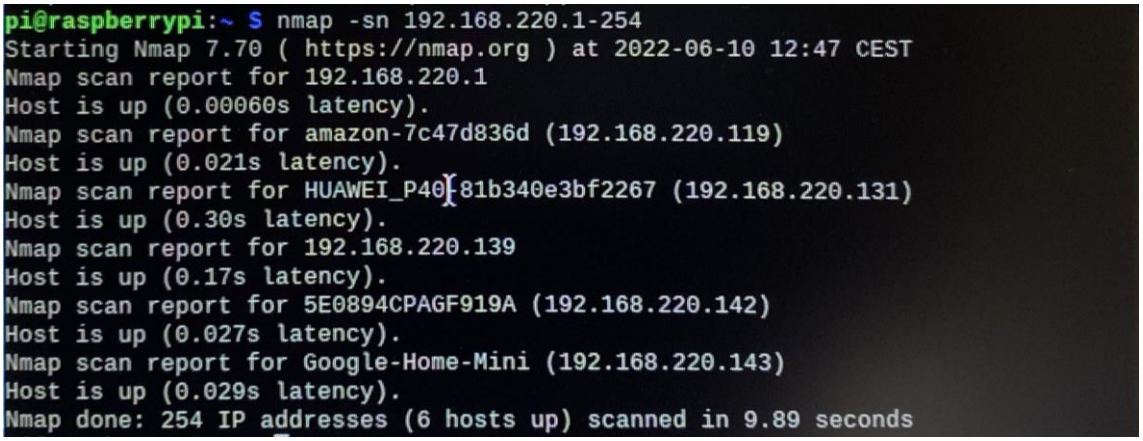


Figura 17. Dispositivos configurados en la red Wi-Fi de la Raspberry Pi, según se indicaba en la arquitectura

3.2.3 Configuración del router

Una vez toda esta arquitectura está montada, tenemos que hacerla accesible al exterior. Según la configuración actual, todo se encuentra en una red privada. Los honeypots están en la red 192.168.1.0/24, por tanto, un atacante exterior no puede acceder a ellos.

Con esta configuración, se ha probado que todos los servicios emulados por el honeypot funcionen correctamente desde un PC en la misma red privada, haciendo Telnet al puerto correspondiente [25].

Sin embargo, no vale con esto. Hay que configurar el router local al que está conectada la Raspberry Pi de manera que el tráfico entrante, sea redirigido a la Raspberry Pi. Esta es la definición de port forwarding o redireccionamiento de puertos, técnica de red que permite el acceso a una persona o equipo externo a una dirección privada dentro de una LAN [26].

Para llevar esto a cabo, accedemos al interfaz de configuración del router que, por defecto, está siempre en la primera dirección IP de la red correspondiente, en este caso 192.168.1.1.

Para hacer un redireccionamiento de puertos, primero tenemos que asegurarnos de que la IP de la Raspberry Pi no vaya a cambiar, pues dejaría de funcionar el redireccionamiento de puertos que hubiésemos hecho. Por eso, aunque por lo general no cambiará la IP de la Raspberry Pi, es mejor asegurarnos que esta nno cambiará, asignándole una IP estática. En la configuración avanzada del router, se puede fijar una IP estática a un dispositivo. Ahora mismo tal y como muestra la Figura 18, la Raspberry Pi tiene la IP 192.168.1.48. Por tanto, asociamos esa IP de forma estática a la dirección MAC del dispositivo, para que no varíe, como se muestra en la Figura 19.

Figura 18. IP dinámica asignada a la Rasperry Pi en la red privada del router local

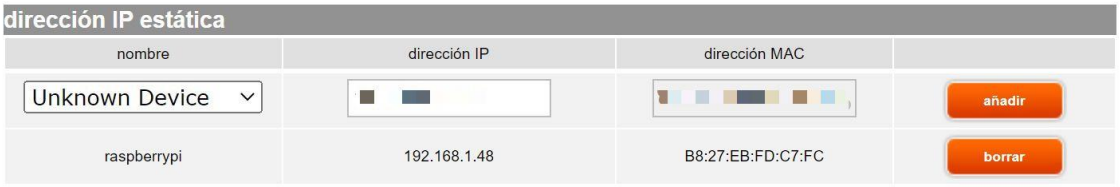
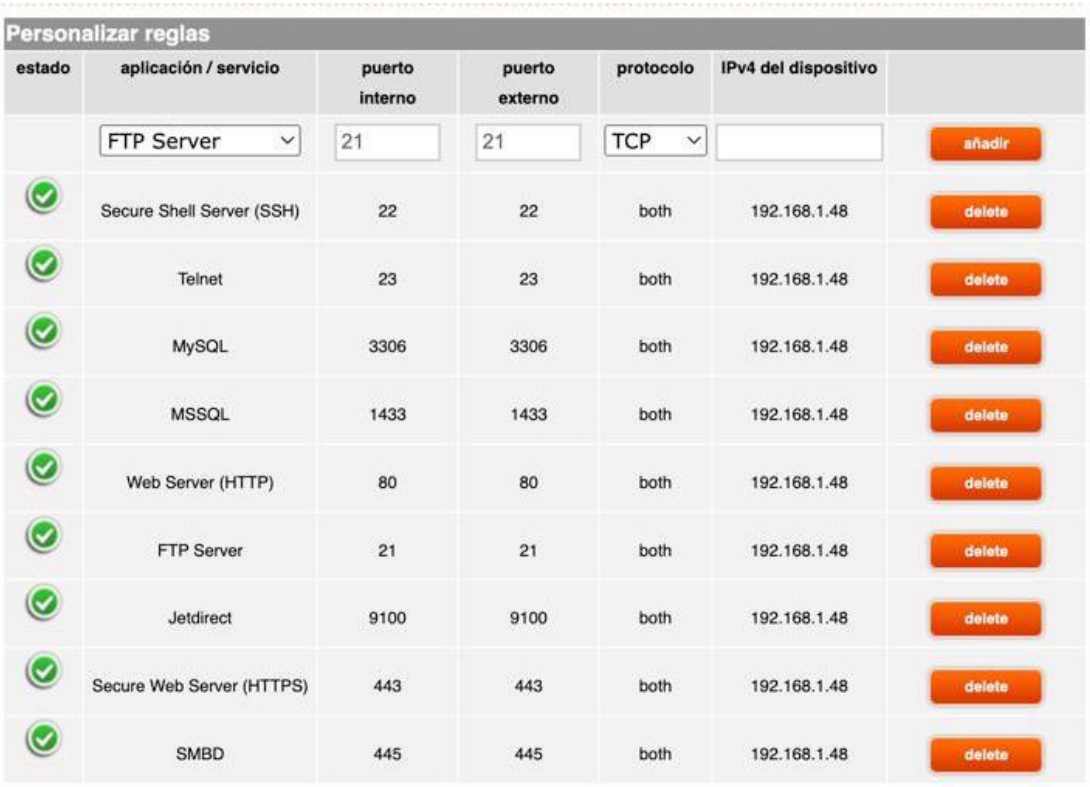


Figura 19. IP estática asignada a la Raspberry Pi

Una vez fijada la IP de la Raspberry Pi, realizamos el redireccionamiento de los puertos en los que hay servicios de los honeypots, para que puedan ser accedidos desde el exterior.

El resultado final de la configuración del redireccionamiento de puertos se muestra en la Figura 20.



Personalizar reglas

estado	aplicación / servicio	puerto interno	puerto externo	protocolo	IPv4 del dispositivo	
	FTP Server	21	21	TCP		añadir
	Secure Shell Server (SSH)	22	22	both	192.168.1.48	delete
	Telnet	23	23	both	192.168.1.48	delete
	MySQL	3306	3306	both	192.168.1.48	delete
	MSSQL	1433	1433	both	192.168.1.48	delete
	Web Server (HTTP)	80	80	both	192.168.1.48	delete
	FTP Server	21	21	both	192.168.1.48	delete
	Jetdirect	9100	9100	both	192.168.1.48	delete
	Secure Web Server (HTTPS)	443	443	both	192.168.1.48	delete
	SMBD	445	445	both	192.168.1.48	delete

Figura 20. Redireccionamiento de puertos en el router

Para comprobar que los puertos son accesibles desde el exterior, se utiliza un smartphone conectado a los datos móviles, obteniendo así una IP externa a la red de la arquitectura, 77.211.4.144.

Consultando desde la Raspberry Pi la dirección IP que tenemos (https://www.whatismyip.com/es/), vemos que es: 37.134.137.249. Son dos IPs públicas de diferentes redes, esto demuestra que, al poder verse entre ellas, cualquier agente externo podrá también.

Por tanto, la arquitectura diseñada, ya está montada. En el siguiente capítulo se procederá a recoger y analizar resultados.

Capítulo 4

4. Recogida y análisis de resultados

En el este capítulo se describen las pruebas realizadas durante la investigación de este proyecto, además de los resultados obtenidos para la arquitectura descrita en el anterior capítulo.

4.1. Recogida de pruebas

A continuación, se presenta una descripción de cómo se han recogido las pruebas de cada uno de los honeypots y motivación de desarrollo de las pruebas, así como el análisis de los resultados obtenidos.

El sistema ha estado operativo en dos periodos de tiempo separados. Por un lado, Cowrie estuvo operativo primero durante un período de un mes a partir del 21 de marzo de 2022. En el caso de Dionaea, si bien estaba operativo durnate este periodo de tiempo, los registros no se almacenaron debidamente. De ahí que se repitiera el proceso en un segundo período de pruebas para recolectar los datos. Dionaea estuvo operativo en su segunda etapa del 21 de mayo de 2022 al 13 de junio de 2022, un total de 24 días.

Para entender los resultados obtenidos, analizaremos ambos honeypots separadamente.

Cabe mencionar que cuando se inició el análisis de los resultados, se recurrió a una herramienta de visualización llamada NtopNG, que descargamos directamente con el comando *sudo apt-get install ntopng*. Para ver el tráfico, se accede a la IP 192.168.220.1, de la Raspberry Pi, en el puerto 3000. El interfaz de esta herramienta, mostrado en la Figura 21, es bastante útil y fácil de interpretar. Sin embargo, muestra resultados en tiempo real, por lo que no nos sirve para los análisis que queremos realizar a lo largo del tiempo. Para eso, es necesario encontrar los registros en los que se almacenan todos los datos de los honeypots.

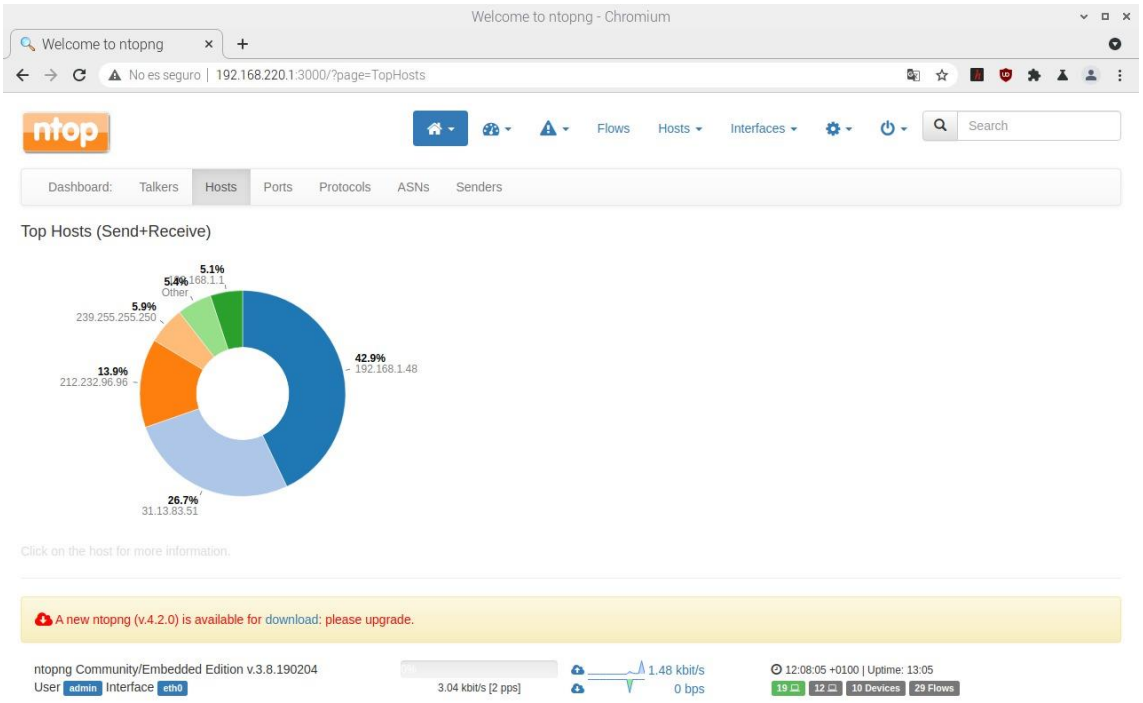


Figura 21. Herramienta de visualización NtopNG

Los resultados de las interacciones con SSH, servicio generado gracias a Cowrie, se almacenan en logs en /var/log/auth.log.X. Se genera una gran

cantidad de logs cada día, por lo que es fácil obtener información a analizar.

Los resultados de Dionaea se encuentran en la carpeta /opt/dionaea/var/lib/dionaea/bistreams. Para cada día se genera una carpeta en la que se registran los logs, como se muestra en la Figura 22. Cada una de estas carpetas tiene una media de 270 logs por día, en total se registraron 6.394 conexiones.

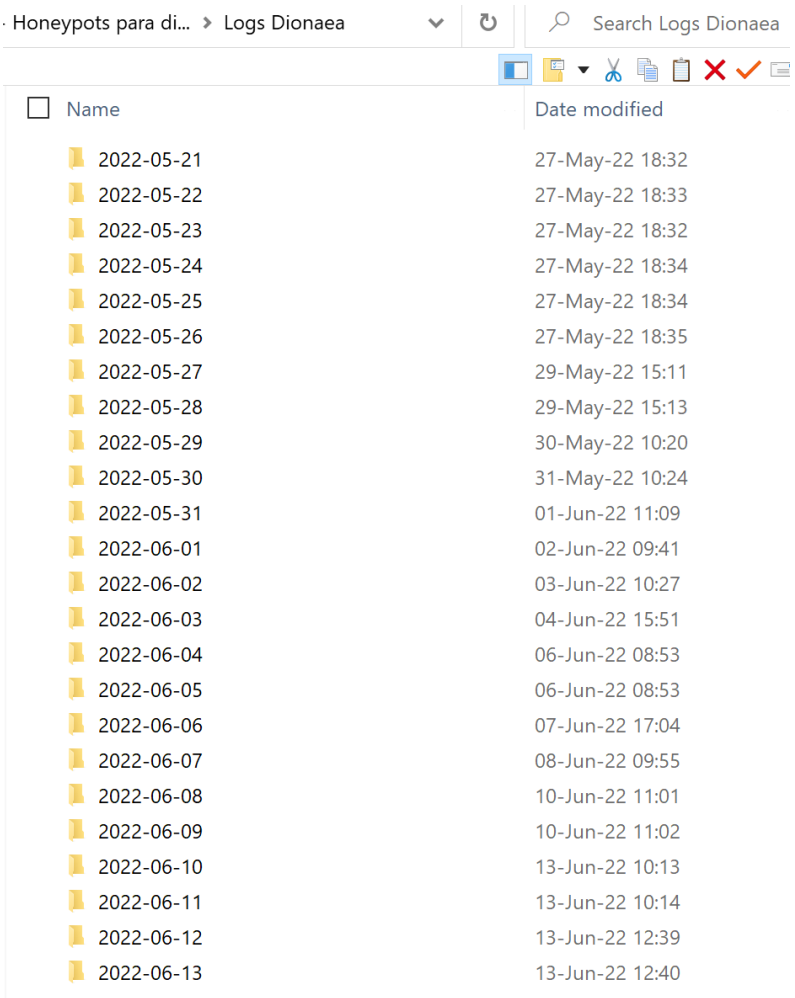


Figura 22. Logs de Dionaea por día

Como se muestra en la figura anterior, estos logs aunque se almacenaban en la Raspberry Pi, se ha utilizado la herramienta Filezilla [27] para pasar los registros a local y que sea más fácil tratar con ellos y analizar sus resultados.

4.2. Análisis de resultados

Una vez tenemos todos los registros organizados y localizados, llevamos a cabo su análisis. A lo largo del proceso de análisis, se ha visto como los honeypots son capaces de leer y extraer información de red, pudiendo extraer conclusiones interesantes sobre los comportamientos, procedimientos y orígenes de los atacantes. Sin embargo, no pueden acceder a actividades maliciosas que pudieran suceder en los dispositivos IoT. Para esto último, utilizaremos Wireshark, que es un analizador de paquetes de red, en la Wi-Fi local de la Raspberry. Esto nos permitirá comprender la seguridad de estos dispositivos y cómo de accesibles o expuestos están a ataques desde el exterior.

4.2.1 Resultados de los honeypots

4.2.1.1 Cowrie

El análisis que vamos a realizar con Cowrie es de los intentos de entrada por SSH. Este protocolo suele ser uno de los más atacados, ya que permite el acceso remoto a los dispositivos.

Después de analizar las direcciones IP, la Figura 23 muestra que los ataques provienen de 15 países y de 26 ciudades diferentes. Aunque China e India son los países que originan un mayor número de ataques, no hay una concentración significativa y se puede concluir que el origen de estos ataques está distribuido por todo el mundo.

Un dato interesante es que no hay atacantes provenientes de España, como se podría haber pensado en un principio, ya que el honeypot se encuentra en Madrid. Aparentemente, los atacantes no se centran en objetivos cercanos, más bien todo lo contrario: tienden a atacar redes remotas.

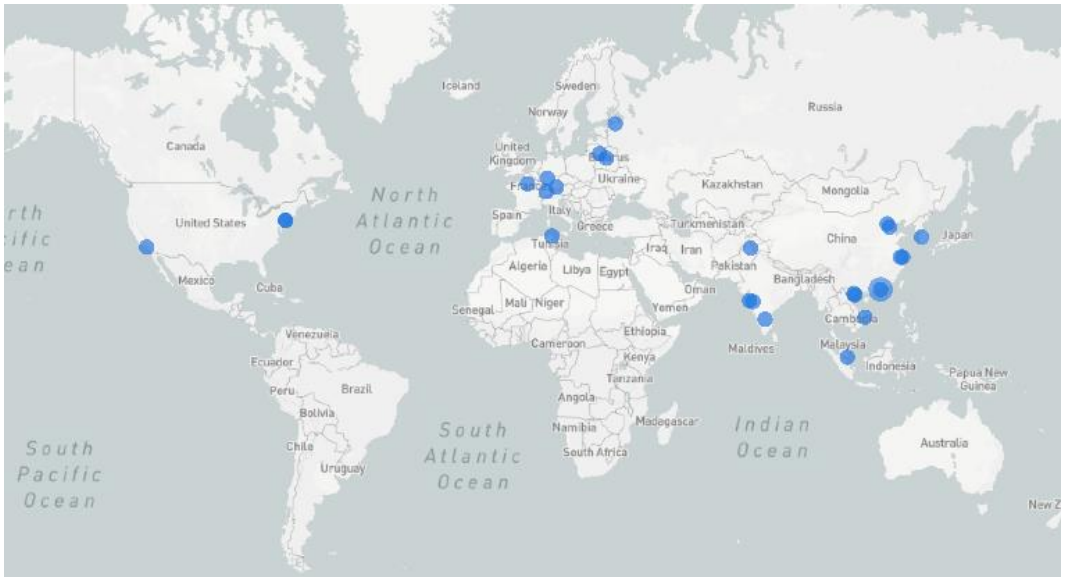


Figura 23. Localización de las IPs de los ataques

Si nos centramos en las 10 IPs más activas, se puede ver en la Figura 24 que estos ataques también tienen orígenes distribuidos por todo el mundo. Estas 10 direcciones IP están localizadas en 8 países de 9 ciudades diferentes y constituyen un 65% de los ataques recibidos.

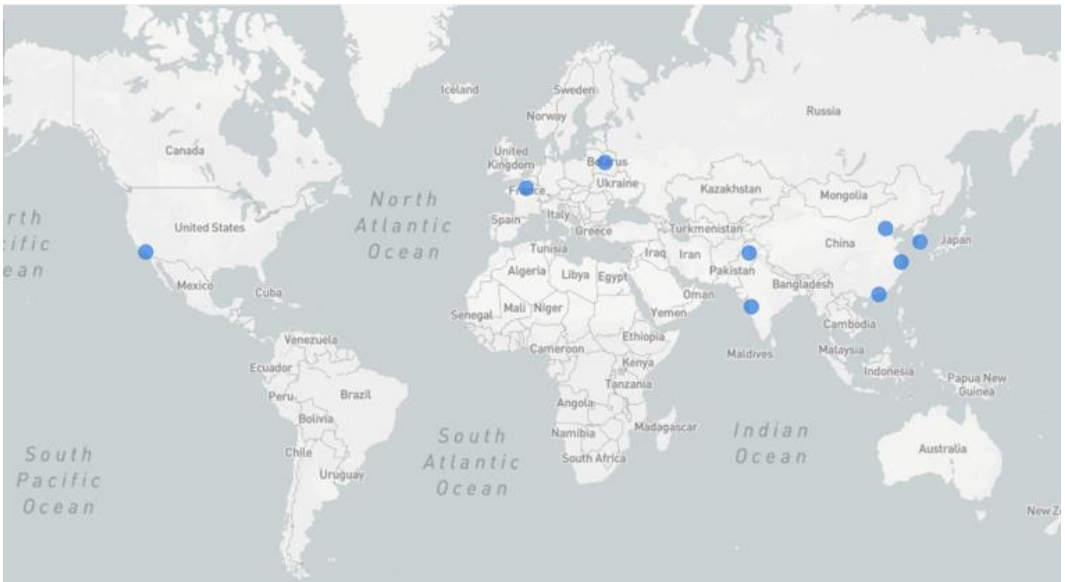


Figura 24. Localización de las 10 IPs más activas

Por otro lado, es interesante destacar los procedimientos con los que estos ataques se llevan a cabo. A modo de ejemplo, en la Figura 25 se puede ver que desde una misma IP (61.177.173.12) se realizan ataques de tres en tres cada cierto tiempo, cambiando de puerto (14724, 13179, 46959, 51270, …). Este es un procedimiento típico cuya intención es no ser detectado y bloqueado por sistemas de detección de intrusiones (IDS), a la vez que se prueban distintos puertos de un mismo equipo para conseguir acceder a alguno de ellos.

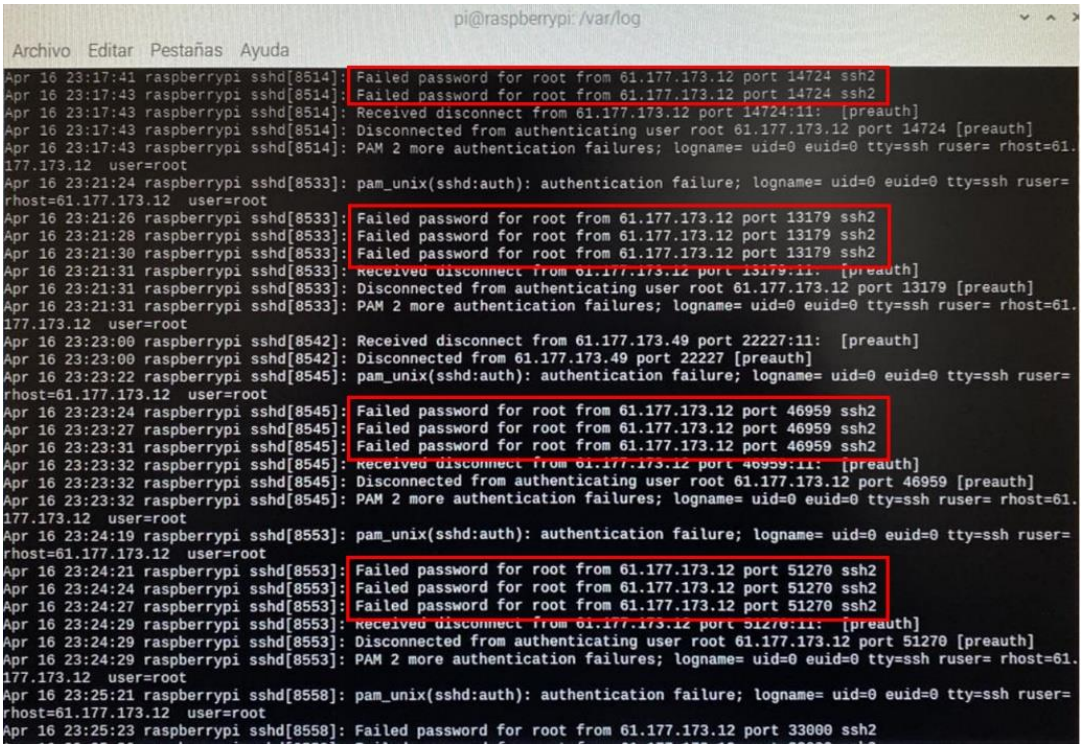


Figura 25. Ejemplo de procedimiento de ataque a SSH

Además, comprobando esa dirección IP en una BBDD de direcciones denunciadas, aparece denunciada 5.061 veces, realizando los mismos ataques a otras víctimas, como muestra la Figura 26.

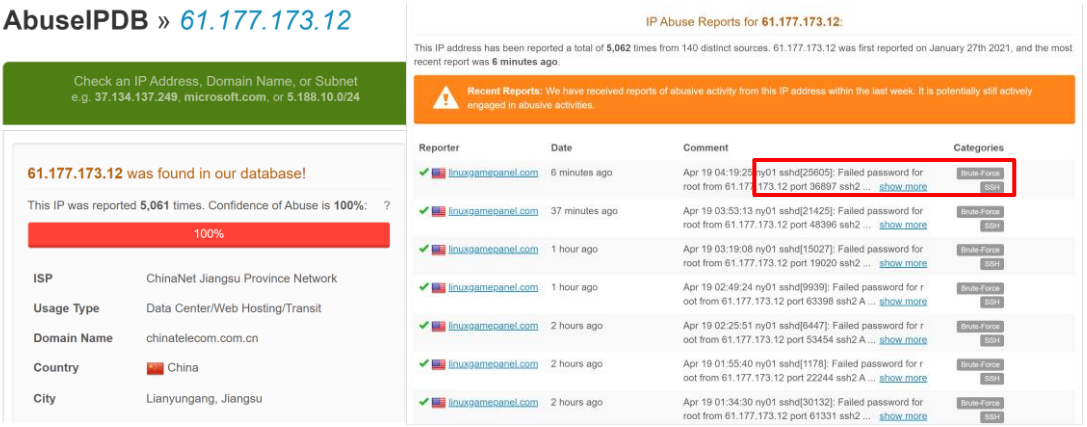


Figura 26. Reporte de abuso de la IP 61.177.173.12

4.2.1.2 Dionaea

Como mencionábamos en el apartado anterior, en los logs de Dionaea hemos obtenido 6.394 conexiones de un total de 1.929 IPs diferentes.

Desde el punto de vista de las IPs, la distribución de los ataques es poco equitativa, esto se debe a que tan solo un 36% de las IPs ha realizado una única conexión, mientras que 1.230 IPs han hecho más de una conexión.

Además, hay algunas IPs que concentran un gran porcentaje de conexiones. En la Tabla V se muestran las 11 IPs con más de 100 conexiones registradas. Estas 11 IPs registran más de un 50% del total de las conexiones. Analizándolas, se ha descubierto algo llamativo: la IP con un mayor número de conexiones (935 conexiones), proveniente de India, no aparece como IP denunciada en una de las BBDD más utilizadas del mercado, AbuseIPDB [28]. Además, otro dato relevante es que las IPs más activas se dedican a atacar siempre al mismo protocolo, ninguna de estas 11 IPs atacan a más de un protocolo, se centran en el indicado en la tabla. Otro dato relevante es que los dos únicos ataques a HTTP son realizados desde Rusia, el resto la mayoría se centran en atacar bases de datos.

En cuanto a las fechas de ataque, de estas 11 IPs, tan sólo 2 han atacado más de un día, el resto realizaron todas las conexiones el mismo día. Esas 2 IPs

coinciden con las que han sido más veces denunciadas: la china 124.133.28.21 y la rusa 193.106.191.48.

Tabla V. IPS MÁS ACTIVAS REGISTRADAS POR DIONAEA

IP	# conexiones	País	Protocolo	¿Denunciada?
117.208.175.104	935	India	MSSQLD	No
41.33.85.50	483	Egipto	MSSQLD	10
52.152.170.42	365	EEUU	FTPD	110
218.2.210.90	257	China	MSSQLD	12
2.155.11.31	219	España	FTPD	No
124.133.28.21	192	China	MYSQLD	6902
183.63.188.226	189	China	MSSQLD	51
182.160.123.98	185	Bangladesh	MSSQLD	61
62.240.106.226	160	Egipto	MSSQLD	65
193.106.191.48	122	Rusia	HTTPD	5755
141.105.66.213	103	Rusia	HTTPD	82

En la Figura 27 aparecen geolocalizadas las 100 IPs más activas. Como en el caso de los ataques a SSH, están distribuidos por todo el mundo, con una gran presencia en China.

Figura 27. Localización de las 100 IPs más activas registradas por Dionaea

Pasando a un enfoque centrado en protocolos, la Tabla VI muestra el porcentaje de ataques dirigido a cada uno de los protocolos analizados y expuestos a Internet emulados por Dionaea.

Existe un protocolo objetivo claro que es la BBDD de MS, MSSQLD, que concentra un 60% de los intentos de conexión, seguido de la transferencia de ficheros, FTP y de HTTP, con un 15% y un 12%, respectivamente.

Tabla VI. PROPORCIÓN DE ATAQUES POR PROTOCOLO

	Nº total conexiones	6394	100%
Por protocolo	MSSQLD	3849	60%
	FTPD	961	15%
	HTTPD	750	12%
	MYSQLD	590	9%
	PRINTERD	138	2%
	SMDB	106	2%

4.2.2 Resultados de Wireshark

Para poder entender bien la seguridad de los dispositivos IoT, primero hay que entender cómo funcionan sus comunicaciones con el exterior. Para ello, utilizaremos Wireshark, un analizador de protocolos, que lee los paquetes que pasan por la red. En este caso, la red será la Wi-Fi local de la raspberry, a la que están conectados los dispositivos IoT.

Arrancamos Wireshark en la Raspberry Pi [29] y vemos el comportamiento, en este caso, de la webcam (192.168.220.142), cuando un agente externo quiere conectarse a ella. Como agente externo, actúa un móvil conectado a los datos móviles, para que tenga una IP fuera de la red de la webcam (47.60.34.222).

El comportamiento de la webcam, sin que nadie intente acceder a ella, se muestra en la Figura 28 antes de la línea azul y es la siguiente:

1. La webcam (192.168.220.142) conecta con el servidor de esta tecnología, localizado en Alemania, con IP 45.43.62.40, preguntando si alguien quiere conectarse a ella. Esto se producirá repetidas veces,

pues la webcam está continuamente buscando a alguien que quiera acceder a ella.

2. La webcam (192.168.220.142) habla con el router (192.168.220.1).
3. Vuelve a conectar con el servidor y después con otro de su red 45.43.62.3 y otros con IP 10.121.182.80 y 10.4.55.202.

En el segundo 53, representado por la línea azul en la figura, el móvil intenta acceder a la webcam. En ese momento, el móvil se comunica con el servidor (45.43.62.40) informándole sobre su intención de comunicarse con el dispositivo. Por eso, en la línea azul, el servidor está comunicando con la webcam, informándole de que el dispositivo móvil quiere comunicar con ella. Algunos paquetes de configuración después, la webcam comunica directamente con el móvil. Una vez la cámara abre esta conexión, el móvil puede comunicar con ella. Pero es la webcam quien tiene que iniciar la conversación, habilitando la comunicación webcam-móvil, después de que el servidor se lo haya indicado. Si no, el móvil o cualquier otro dispositivo externo, no podrá acceder a ella, pues la conexión directa está protegida por el router.

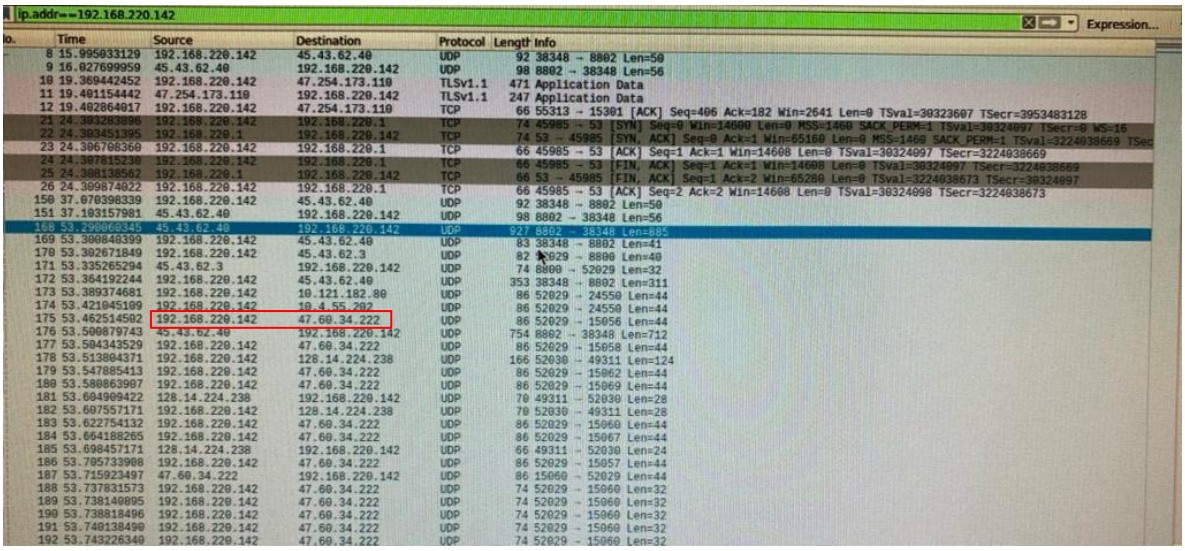


Figura 28. Captura con Wireshark de las comunicaciones de la webcam

Esto demuestra que los dispositivos IoT son suficientemente seguros, ya que pasan siempre por el servidor antes de iniciar la comunicación con otros dispositivos. Lo mismo ocurre con la comunicación de los SPAs y de los wearables, aunque en este último caso, es el smartphone el que se conecta con el servidor de la aplicación correspondiente dependiendo del wearable que sea. Siempre la comunicación la habilita el dispositivo IoT, después de que el servidor le haya indicado que un agente externo quiere comunicarse con él, por tanto, los dispositivos están protegidos.

Sin embargo, si existiera un redireccionamiento de puertos interno en la Raspberry Pi, como ocurre en el router local, un atacante podría acceder directamente a la cámara. Para comprobar este escenario, realizamos un port forwarding entre las redes de las interfaces eth0 y wlan0 de la Raspberry Pi [30].

Una vez la webcam es accesible desde exterior, si cualquier agente externo intenta acceder a la IP pública del router, 37.134.137.249, en el puerto 80 se encontrará con el interfaz de configuración de la webcam, y, por tanto, podrá acceder a su contenido de video también, ya que es uno de los servicios que se ofrece en el interfaz de configuración, puesto que se está haciendo una redirección de puertos primero a 192.168.1.48 (eth0) y seguidamente a 192.168.220.142 (wlan0).

Dejando Wireshark analizando tráfico durante la noche del 04 de junio de 2022, al día siguiente comprobamos que ha habido varios ataques a la webcam.

En la Figura 29 se muestra un ejemplo de ataque a la cámara desde la IP 111.90.145.4 que intenta acceder a diferentes páginas y bases de datos, haciendo uso del protocolo HTTP, además genera mucho tráfico TCP, que es el protocolo utilizado en la comunicación entre un dispositivo externo y la webcam.

Time	Source	Destination	Protocol	Length	Info
14059.36586	111.90.145.4	192.168.220.142	HTTP	360	GET /hudson/script HTTP/1.1
14059.36731	192.168.220.142	111.90.145.4	TCP	54	80 > 63970 [ACK] Seq=1 Ack=307 Win=15680 Len=0
14059.36768	192.168.220.142	111.90.145.4	TCP	54	80 > 63922 [FIN, ACK] Seq=139 Ack=301 Win=15680 Len=0
14059.36955	192.168.220.142	111.90.145.4	HTTP	192	HTTP/1.1 404 Not Found (text/html)
14059.53026	111.90.145.4	192.168.220.142	TCP	54	63922 > 80 [ACK] Seq=301 Ack=140 Win=131072 Len=0
14059.53369	111.90.145.4	192.168.220.142	TCP	54	63970 > 80 [FIN, ACK] Seq=307 Ack=139 Win=131072 Len=0
14059.53396	111.90.145.4	192.168.220.142	TCP	66	63992 > 80 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MSS=1436 WS=256 SACK_PERM=1
14059.53558	192.168.220.142	111.90.145.4	TCP	66	80 > 63992 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=16
14059.56994	192.168.220.142	111.90.145.4	TCP	54	80 > 63970 [ACK] Seq=139 Ack=308 Win=15680 Len=0
14059.59069	192.168.220.142	111.90.145.4	TCP	54	80 > 63970 [FIN, ACK] Seq=139 Ack=308 Win=15680 Len=0
14059.69835	111.90.145.4	192.168.220.142	TCP	54	63992 > 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
14059.6985	111.90.145.4	192.168.220.142	HTTP	353	GET /script HTTP/1.1
14059.70055	192.168.220.142	111.90.145.4	TCP	54	80 > 63992 [ACK] Seq=1 Ack=300 Win=15680 Len=0
14059.70209	192.168.220.142	111.90.145.4	HTTP	192	HTTP/1.1 404 Not Found (text/html)
14059.75332	111.90.145.4	192.168.220.142	TCP	54	63970 > 80 [ACK] Seq=308 Ack=140 Win=131072 Len=0
14059.86657	111.90.145.4	192.168.220.142	TCP	54	63992 > 80 [FIN, ACK] Seq=300 Ack=139 Win=131072 Len=0
14059.89995	192.168.220.142	111.90.145.4	TCP	54	80 > 63992 [ACK] Seq=139 Ack=301 Win=15680 Len=0
14059.93653	111.90.145.4	192.168.220.142	TCP	66	64019 > 80 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MSS=1436 WS=256 SACK_PERM=1
14059.93811	192.168.220.142	111.90.145.4	TCP	66	80 > 64019 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=16
14060.10043	111.90.145.4	192.168.220.142	TCP	54	64019 > 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
14060.10048	111.90.145.4	192.168.220.142	HTTP	362	GET /sqlite/main.php HTTP/1.1
14060.1024	192.168.220.142	111.90.145.4	TCP	54	80 > 64019 [ACK] Seq=1 Ack=309 Win=15680 Len=0
14060.10287	192.168.220.142	111.90.145.4	TCP	54	80 > 63992 [FIN, ACK] Seq=139 Ack=301 Win=15680 Len=0
14060.10489	192.168.220.142	111.90.145.4	HTTP	192	HTTP/1.1 404 Not Found (text/html)
14060.26527	111.90.145.4	192.168.220.142	TCP	54	63992 > 80 [ACK] Seq=301 Ack=140 Win=131072 Len=0

Figura 29. Captura de Wireshark mostrando el acceso desde 111.90.145.4

Comprobando esta IP en la herramienta AbuseIPDB, previamente mencionada, en la Figura 30 vemos que ya ha sido reportada 12 veces previamente, por lo que confirmamos que ha sido un ataque malicioso, que proviene de Malasia.

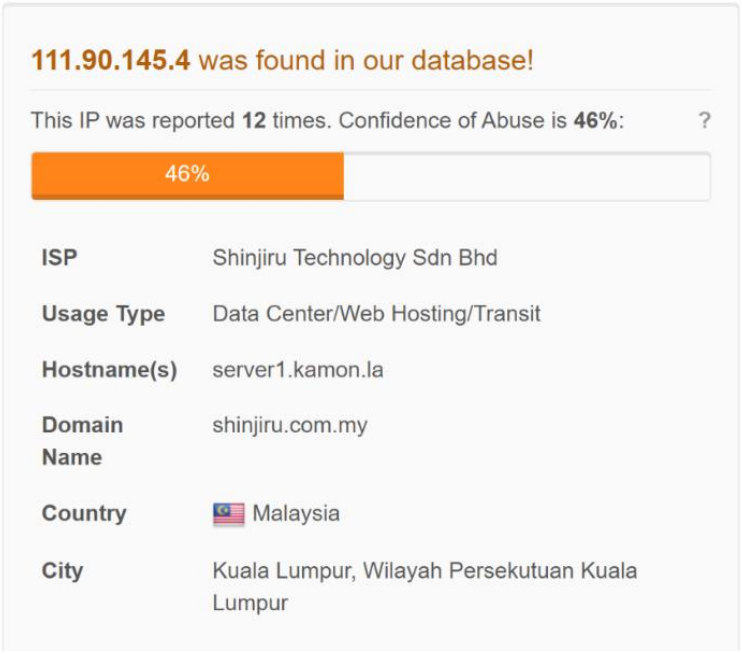


Figura 30. Reporte de la IP 111.90.145.4

Tras haber analizado todos estos datos, podemos hacernos una mejor idea sobre el entorno de seguridad de los dispositivos IoT, así como de la manera de actuar de los atacantes y sus objetivos primordiales.

En el siguiente capítulo, se expondrán las principales conclusiones del trabajo junto con líneas de desarrollo futuro como posibilidades de continuar esta investigación.

Capítulo 5

5. Conclusiones y trabajos futuros

En este capítulo hablaremos sobre las conclusiones extraídas tras la realización del proyecto, además de las posibles vías de trabajo que se podrían plantear. Para esto, la mejor forma de sacar conclusiones es analizar los objetivos establecidos en el Capítulo 1 y ver si se han cumplido.

5.1. Conclusiones

Los objetivos planteados en un inicio eran:

- Análisis de honeypots: después de analizar varios de ellos, se llegó a un mejor conocimiento del estado del arte necesario como para poder elegir correctamente en los apartados posteriores. Se estableció una tipología que permite clasificar la multitud de diferentes honeypots existentes, según sus diferentes características (escalabilidad, nivel de interacción, disponibilidad del código fuente, objetivo…). Esta tipología sirve para escoger entre los diferentes honeypots, en base a las necesidades que tenga el proyecto. También, se analizaron otras investigaciones y se contrastó con ellas la forma de elección de los honeypots implementados.

- Análisis de dispositivos IoT: igual que con los honeypots, se llevó a cabo un análisis de tres diferentes tipos de dispositivos IoT (webcam, wearables y SPAs), y la forma de conexión de cada una de ellas, para poder después conocer cómo funciona su seguridad. Además, se explicaron las amenazas y vulnerabilidades que suelen afectar a este tipo de dispositivos.
- Elección de honeypots y dispositivos IoT: una vez se conocían los diferentes elementos, se escogieron varios para la implementación de un sistema. Esto se desarrolló de forma correcta, dando lugar a la fase de diseño.
- Diseño de la arquitectura: este objetivo se cumplió exitosamente al pensar en un sistema que implementase todos los elementos escogidos en la fase anterior y cómo conectarían unos con otros.
- Implementación del sistema: esta fase se llevó a cabo con éxito siguiendo las indicaciones del punto anterior. Sin embargo, al llevar a cabo esta implementación se ha comprobado que, al tener dos tarjetas en la Rasberry Pi, cada una en una red, los honeypots están implementados en la red 192.168.1.0/24 y son accesibles desde el exterior de la red gracias al redireccionamiento de puertos en el router local, pero los dispositivos están en la red 192.168.220.0/24. Por tanto, por un lado, se obtiene información del comportamiento de los atacantes contra los protocolos típicos de dispositivos IoT (SSH, HTTP, FTP, …) gracias a la información recolectada por los honeypots y por otro lado se ha visto el tráfico interno de los dispositivos IoT, permitiéndonos saber cómo un atacante puede acceder a ellos. Las conclusiones de ambas partes se extrajeron en el último de los objetivos del proyecto, desarrollado a continuación.
- Recolección y análisis de los resultados obtenidos: para llevar a cabo el análisis de la información recolectada, en un primer momento se quiso hacer uso de herramientas gráficas como ELK (Elasticsearch, Logstash y Kibana), stack opensource para filtrar, almacenar y visualizar logs, y DionaeaFR. Estas herramientas tomarían los registros generados por los honeypots y sacarían

conclusiones mediante gráficos, sin necesidad de mucho análisis manual. Sin embargo, DionaeaFR lleva sin actualizarse desde 2014 [31] y aunque existen referencias de su uso posteriores, no hay nada posterior a 2017. Al proceder a su instalación se dieron errores de instalación que hacían imposible utilizarlo. Por otro lado, ELK generó errores de firmas en su instalación que no permitieron su implementación. Por eso, se ha realizado un análisis más manual, con todos los ficheros de registro, del que se han extraído las siguientes conclusiones:

1. Por el número de ataques registrados en los honeypots y número de IPs de las que proceden, vemos que la mayoría de ataques son intentos de denegación de servicio o acceso por fuerza bruta, en el caso de los login de SSH, de forma separada en el tiempo para evitar ser detectados por los IDSs.
2. Los protocolos relacionados con bases de datos son objetivos preferidos por los atacantes.
3. Los ataques están realizados desde partes muy diversas del mundo, no se concentran de forma relevante en un país en concreto.
4. Los dispositivos IoT son suficientemente seguros, ya que las conexiones con el exterior se inician a través de los servidores propietarios. A no ser que se consiguiera una redirección del tráfico directa en el router para acceder a estos sin pasar por sus servidores, ataque que sería más complejo de realizar, los dispositivos están configurados de forma segura.

Sin embargo, de todas estas conclusiones podríamos extraer más información en diferentes líneas de trabajo futuras.

5.2. Trabajos futuros

Como último apartado de este capítulo, se ofrece una descripción de varias líneas de desarrollo futuro como posibilidades a la hora de continuar con el trabajo realizado en el proyecto.

5.2.1 Análisis de las vulnerabilidades en la comunicación entre dispositivo IoT y servidor

Estudio de las posibles vulnerabilidades en la comunicación entre los servidores y los dispositivos IoT. Dado que el establecimiento de comunicación entre un dispositivo IoT y cualquier otro agente suele pasar primero por el servidor correspondiente, una línea de investigación interesante sería ver cómo funciona este proceso y cómo un atacante podría interferir en él para realizar un ataque. Es decir, ver qué ataques pueden sufrir esas aplicaciones.

5.2.2 Desarrollo de los honeypots

Los honeypots desarrollados simulan dispositivos IoT, sin embargo, una de las líneas de desarrollo podría ser mejorar esas simulaciones para que el sistema parezca más real y atraiga más a los atacantes. De esa manera los atacantes podrían realizar más acciones y, por tanto, podríamos estudiar más allá en su comportamiento.

Bibliografía

[1] L. S. Vailshery, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030," 2022. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.

[2] J. Armiñana Gorriz, "Seguridad en Internet de las Cosas Honeypot to capture IoT-attack methods," pp. 1–68, 2018, [Online]. Available: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/82136/6/parriagaTFM0618memoria.pdf.

[3] A. Acién Gómez, "Análisis de vulnerabilidades en IoT para el despliegue de honeypots," *Esc. Técnica Super. Ing. Informática*, p. 54, 2018.

[4] "RAYUELA." https://www.rayuela-h2020.eu/about-us/.

[5] "RAYUELA logo." https://www.rayuela-h2020.eu/wp-content/uploads/2021/01/rayuela_logo01.png.

[6] O. Espinosa, "Qué es y para qué sirve un Honeypot," *RedesZone*, 2021, [Online]. Available: https://www.redeszone.net/tutoriales/seguridad/que-es-honeypot/.

[7] F. C. Sheng, "Network Isolation and Security Using Honeypot," no. May, 2019.

[8] L. Spitzner, "To Build A Honeypot." 1999, [Online]. Available: http://www.spitzner.net/honeypot.html.

[9] J. M. González Aparicio and J. J. Jimenez Copete, "¿Qué es un honeypot?" https://candytraps.wordpress.com/que-es-un-honeypot/ (accessed Jun. 06, 2022).

[10] W. Fan, Z. Du, D. Fernandez, and V. A. Villagra, "Enabling an Anatomic View to Investigate Honeypot Systems: A Survey," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3906–3919, 2018, doi: 10.1109/JSYST.2017.2762161.

[11] S. E. Luis, T. José, and E. López, "Monitorización Y Prevención De Ataques," 2017.

[12] "El verdadero significado del término daemon." https://blog.desdelinux.net/el-verdadero-significado-del-termino-daemon/ (accessed Jun. 07, 2022).

[13] K. Asthon, "That ' Internet of Things ' Thing," *RFiD J.*, p. 4986, 2009, [Online]. Available: www.itrco.jp/libraries/RFIDjournal-That Internet of Things Thing.pdf.

[14] M. (Deloitte) Gracia, "IoT - Internet Of Things." [Online]. Available: https://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html.

[15] W. Wobcke, A. Nguyen, H. Van Ho, and A. Kzywicki, "The Smart Personal Assistant: An

Overview."

[16] C. V. Amores, "Security and Privacy Analysis of Smart Personal Assistants Available on the Market Background . Security in Smart," 2021.

[17] J. F. de la Fuente, "Análisis de dispositivos wearable para menores desde un punto de vista de privacidad y seguridad," *Nuevos Sist. Comun. e Inf.*, pp. 2013–2015, 2021.

[18] J. Elio, "La historia de los wearables: cinco siglos intentando vestir tecnología," 2016.

[19] M. Wedd, "Bluetooth IoT Applications: From BLE to Mesh," *IoT For All*, 2020.

[20] O. Quiñonez Muñoz, *Internet de las Cosas (IoT)*. 2019.

[21] J. Fruhlinger, "The Mirai botnet explained: How teen scammers and CCTV cameras almost brought down the internet," *CSO*, 2018.

[22] J. Jiménez, "Ataques más comúnes a los dispositivos IoT," *RedesZone*, 2020.

[23] GitHub, "Dionaea Installation." https://dionaea.readthedocs.io/en/latest/installation.html.

[24] EVAL2A, "Honeypot part 1: Setting up Cowrie and Dionaea." https://eval2a.wordpress.com/2017/12/04/honeypot-part-1-setting-up-cowrie-and-dionaea/.

[25] Y. Fernández, "Telnet: qué es y cómo activarlo en Windows 10," 2020. https://www.xataka.com/basics/telnet-que-como-activarlo-windows-10.

[26] A. Crespo, "¿Qué es port forwarding y por qué debo bloquear los puertos de mis dispositivos?," *RedesZone*, 2017.

[27] T. Kosse, "FileZilla - The free FTP solution." [Online]. Available: https://filezilla-project.org/.

[28] "AbuseIPDB." abuseipdb.com.

[29] B. Mayes, "Project: Installing Wireshark on Raspberry Pi." https://unboxing-tomorrow.com/project-installing-wireshark-on-raspberry-pi/.

[30] "Bridge or port forward Ethernet to Wi-Fi," 2018. https://forums.raspberrypi.com/viewtopic.php?t=221871.

[31] R. Espadas, "DionaeaFR." https://github.com/rubenespadas/DionaeaFR.

Anexo A

Presupuesto

La Tabla A.1 muestra el presupuesto del proyecto desarrollado para el TFM descrito en esta memoria.

Tabla A.1: Presupuesto económico del Trabajo Fin de Máster

Costes		Horas	Precio/Hora	Total
Coste de mano de obra (Costes directos)		450	35 €	**15750 €**
Coste de recursos materiales (Costes directos)	Precio de compra	Uso en meses	Amortización en años	Total
Ordenador personal	2699 €	7	5	**314,88 €**
Total				**16064,88 €**
Gastos generales (Costes indirectos)	15 % sobre Costes Directos			**2409,73 €**
Beneficio industrial	6 % sobre (Costes Directos + Costes Indirectos)			**1108,48 €**
Total				**19583,09 €**
Material fungible				
Dahua Vandal Proof Wi-Fi Dome Camera				100 €
Huawei P40				499 €
Google Home Mini				59,99 €
Apple HomePod Mini				99 €
Amazon Echo Show 5				84,99 €
vívofit jr. 2				64,99 €
Huawei Honor Band 5				32,99 €
Huawei Honor Watch Es				79,90 €
Mi Band 5				29,99 €
Fitbit Ace 3				79,95 €
TOOBUR Smartwacth				39,99 €
Raspberry Pi 3				50 €
Total				**1220,79 €**
Subtotal presupuesto				**20803,88 €**
IVA aplicable	21 %			4160,78 €
Total presupuesto				**24.964,66 €**

Anexo B

Objetivos de Desarrollo Sostenible

La seguridad en los sistemas es hoy en día un requisito fundamental para el funcionamiento de las empresas y la creación de una sociedad con cimientos sólidos.

Hoy en día, los conflictos entre países, se dan más en el espacio ciber que en físico. Por ello, una seguridad fuerte en los sistemas, defiende el establecimiento de paz entre naciones y evita que se den conflictos de este tipo.

Por esto, se considera que este Trabajo de Fin de Máster se alinea con los Objetivo de Desarrollo Sostenible número nueve y dieciséis: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación (9) y Paz, justicia e instituciones sólidas (16).

Este proyecto pretende dar a conocer la seguridad en el entorno IoT y sus vulnerabilidades y principales ataques para poder evitarlos.