Universidad Internacional de La Rioja (UNIR)

ESIT

Máster universitario en Seguridad Informática

Evaluación de actualizaciones y parches de seguridad que recibe la tecnología bluetooth en los dispositivos móviles Android.

Trabajo Fin de Máster

presentado por: Paternina León, Alberto Manuel

Director/a: Martínez Herraiz, José Javier

Ciudad: Montería
Fecha: 27/01/2020

Tabla de Contenido

1 Resumen....9
2 Abstract....10
3 Introducción....11
3.1 Motivación....11
3.2 Planteamiento del problema....11
3.3 Estructura del trabajo....12
3.3.1 Capítulo 2. Estado del arte....12
3.3.2 Capítulo 3. Objetivos y metodología de trabajo....12
3.3.3 Capítulo 4. Desarrollo específico de la contribución....12
3.3.4 Capítulo 5. Conclusiones y trabajos futuros....13
3.3.5 Capítulo 6. Referencias....13
3.3.6 Capítulo 7. Anexos....13
4 Estado del arte....14
4.1 Marco teórico....14
4.1.1 Descripción De La Tecnología Bluetooth....14
4.1.2 Desarrollo De La Tecnología Bluetooth....14
4.1.3 Vulnerabilidad....15
4.1.4 Base de Datos de Vulnerabilidad Nacional (NVD).....15
4.1.5 Sistema de puntuación de vulnerabilidad común (CVSS)....16
4.1.6 Proyecto de Código Abierto Android (AOSP)....17
4.1.7 Pentesting....17
4.1.8 Tipos de Pentesting....17
4.1.9 Singer Board Computer (SBC)....17
4.1.10 Kali Linux....18

4.1.11 Bring Your Own Devices (BYOD) 18
4.2 Antecedentes 19
4.2.1 Bluetooth V1 19
4.2.2 Bluetooth V2.0 + EDR 19
4.2.3 Bluetooth V2.1 + EDR 20
4.2.4 Bluetooth V3.0 20
4.2.5 Bluetooth V4.0 20
4.2.6 Bluetooth Versión 4.1 21
4.2.7 Bluetooth Versión 4.2 21
4.3 Estado actual 21
4.3.1 Bluetooth V5.0 21
4.3.2 Vulnerabilidades bluetooth 22
4.4 Trabajos relacionados 22
4.4.1 BlueBorne 22
4.4.2 Bluetooth Vulnerability Finder: 23
5 Objetivos y Metodología del trabajo 24
5.1 Objetivo general 24
5.2 Objetivos específicos 24
5.3 Metodología 24
6 Desarrollo especifico de la contribución 26
6.1 Descripción del piloto 26
6.2 Preparación del ambiente base 27
6.3 Desarrollo del piloto 28
6.3.1 Clasificación de vulnerabilidades bluetooth y Patch de seguridad. 29
6.3.2 Identificación de vulnerabilidades en dispositivos móviles android. 32
6.3.3 Instalación de las actualizaciones de seguridad. 38
6.3.4 Validación de las actualizaciones de seguridad. 40
6.3.5 Informe de los resultados 44
7 Conclusiones y trabajos futuros 46

7.1 Trabajos futuros 47
8 Referentes bibliográficos 48
9 Anexos 51
9.1 Anexo 1: Instalación de Kali Linux en Raspberry Pi 4 modelo B.....51
9.2 Anexo2: Instalación y manual de uso de la herramienta BlueExploitApp. 54
9.3 Anexo3: Habilitar la opción de depuración USB. 58

Índice de tablas

Tabla 1 Posicionamiento Global de Marcas de celulares Inteligentes 3Q 2019. 26
Tabla 2 Equipos de Cómputo Empleados en el Piloto. 26
Tabla 3 Dispositivos Móviles Empleados en el Piloto. 27
Tabla 4 Vulnerabilidades Bluetooth en Dispositivos Android 29
Tabla 5 Vulnerabilidades Bluetooth con Puntaje Crítico y Alto que Afectan Dispositivos Android.....30
Tabla 6. Fecha de lanzamiento de los Patch de seguridad emitidos por AOSP y algunos fabricantes de dispositivos móviles android a nivel global. 31

Índice de ilustraciones

Ilustración 1. Métricas del sistema de puntuación CVSS v2 (Fuente: https://www.incibe-cert.es/sites/default/files/blog/cvss3-0/metricasv2.png) 16
Ilustración 2. Métricas del sistema de puntuación CVSS v3 (Fuente: https://www.incibe-cert.es/sites/default/files/blog/cvss3-0/grupos_metricas_cvss3.png) 16
Ilustración 3. Impresión 3D Caja Raspberry Pi con Adaptación de Pantalla Lcd, (Fuente: Elaboración Propia).....27
Ilustración 4. Dispositivo Portable Para la Identificación y Explotación de Vulnerabilidades Bluetooth, (Fuente: Elaboración Propia).....28
Ilustración 5. Metodología Para la Identificación de Vulnerabilidades Bluetooth y Validación de los Patch de Seguridad, Dispuesto Para los Dispositivos Móviles Android. (Fuente: Elaboración Propia.).....28
Ilustración 6. Resultado de la Búsqueda de Vulnerabilidades Bluetooth con Puntuación de Gravedad Alta. (Fuente: Elaboración Propia). 29
Ilustración 7. Proceso Para la Identificación de Vulnerabilidades, Validación de Actualizaciones de Seguridad en Dispositivos Móviles y creación de Whitelist y Blacklist. (Fuente: Elaboración Propia).....32
Ilustración 8. Conexión del Dispositivo 1 (LG-K430) y el Dispositivo Portable (Raspberry Pi 4) (Fuente: Elaboración Propia).....33
Ilustración 9. Dispositivos Conectados con la Aplicación BlueExploitApp (Fuente: Elaboración Propia). 33
Ilustración 10. Implementación de la Función search_devices() (Fuente: Elaboración Propia). 34
Ilustración 11. Implementación de la Función selected_devices() (Fuente: Elaboración Propia). 34
Ilustración 12. Ventana Para Agregar los Dispositivos a la Base de Datos de la Aplicación BlueExploitApp (Fuente: Elaboración Propia).....35
Ilustración 13. Implementación de la Función btnAddDevices() (Fuente: Elaboración Propia). 35
Ilustración 14. Botón analizar habilitado para iniciar la identificación de vulnerabilidades y listar las actualizaciones de seguridad disponibles (Fuente: Elaboración Propia). 36
Ilustración 15. Implementación de la Función analizeDevices() (Fuente: Elaboración Propia). 36
Ilustración 17. Resultado del Análisis de Vulnerabilidades en el Dispositivo Móvil LG-K430 (Fuente: Elaboración Propia).....37

Ilustración 18. Conexión del Dispositivo 2 (Moto G[5]) y el Dispositivo Portable (Raspberry Pi 4) (Fuente: Elaboración Propia)....37
Ilustración 19. Resultado del análisis del dispositivo 2 (Moto G[5]) (Fuente: Elaboración Propia). 37
Ilustración 20. Instalación de Actualizaciones de Seguridad en el Dispositivo Móvil LG-K430 (Fuente: Elaboración Propia)....39
Ilustración 21. Ultima Actualización de Seguridad Disponible para el Dispositivo Móvil Moto G[5] (Fuente: Elaboración Propia)....40
Ilustración 22. Descarga del Exploit que Demuestra la Existencia de la Vulnerabilidad CVE-2017-0785 (Fuente: Elaboración Propia). 41
Ilustración 23. Verificación y Encendido del Adaptador Bluetooth en la Máquina Atacante. (Fuente: Elaboración Propia)....41
Ilustración 24. Lista de Direcciones MAC de los Dispositivos Móviles con el Adaptador Bluetooth Encendido (Fuente: Elaboración Propia). 42
Ilustración 25. Resultado de la Ejecución del Exploit CVE-2017-0785 en el Dispositivo 1 (Fuente: Elaboración Propia)....42
Ilustración 26. Resultado de la Ejecución del Exploit CVE-2017-0785 en el Dispositivo 2 (Fuente: Elaboración Propia)....43
Ilustración 27. Resultado de la Ejecución del Exploit CVE-2017-0785 en el Dispositivo 1 (Fuente: Elaboración Propia)....44
Ilustración 28. Resultado de la Ejecución del Exploit CVE-2017-0785 en el Dispositivo 2 (Fuente: Elaboración Propia)....44
Ilustración 29. Pasos Para Descargar la Imagen de Kali Linux (Fuente: Elaboración Propia). 51
Ilustración 30. Instalación del Programa Win32diskimager en Windows 10 (Fuente: Elaboración Propia)....51
Ilustración 31. Pasos Para Copiar la Imagen y Crear el Arranque de Kali Linux en la Memoria SSD (Fuente: Elaboración Propia). 52
Ilustración 32. Actualización del Sistema Operativo Kali Linux (Fuente: Elaboración Propia). 52
Ilustración 33. Secuencia de Pasos Para la Instalación de los Controladores de un Display Touch Screen de 3.2 Pulgadas en una Raspberry pi 4 Con Sistema Operativo Kali Linux (Fuente: Elaboración Propia)....53
Ilustración 34. Clonación del Repositorio de la Herramienta BlueExploitApp utilizando la Terminal (Fuente: Elaboración Propia)....54

Ilustración 35. Descarga del Repositorio de la Herramienta BlueExploitApp (Fuente: Elaboración Propia).....54
Ilustración 36. Instalación de las Librerías Necesarias para Ejecutar la Herramienta BlueExploitApp y la Configuración del Servidor de Base de Datos (Fuente: Elaboración Propia) 55
Ilustración 37. Instalación de la Base de Datos de la Herramienta BlueExploitApp (Fuente: Elaboración Propia).....55
Ilustración 38. Puesta en Marcha de la Herramienta BlueExploitApp y Ruta de Acceso (Fuente: Elaboración Propia).....56
Ilustración 39. Página Principal de la Herramienta BlueExploitApp (Fuente: Elaboración Propia) 56
Ilustración 40. Lista de dispositivos disponibles para ser analizados (Fuente: Elaboración Propia) 56
Ilustración 41. Ventana Versiones de Android y Lista de versiones de Android (Fuente: Elaboración Propia).....57
Ilustración 42. Ventana Vulnerabilidades y Lista de Vulnerabilidades Bluetooth que Afectan a los Dispositivos Android (Fuente: Elaboración Propia) 57
Ilustración 43. Ventana Actualizaciones y Lista de Actualizaciones Disponibles por Versión de Android y Fabricante (Fuente: Elaboración Propia) 58
Ilustración 44. Ventana Exploit y Lista Exploit que Evidencia la Vulnerabilidad Detectada. (Fuente: Elaboración Propia).....58
Ilustración 45. Habilitando la Opción de Programador en Android (Fuente: Elaboración propia) 58
Ilustración 46. Habilitando la Opción de Depuración USB en Android (Fuente: Elaboración Propia) 59

1 Resumen

En la actualidad los dispositivos móviles son considerados una herramienta indispensable, para el desarrollo de actividades personales, escolares y laborales. Es por esto, que muchas organizaciones están implementando políticas como BYOD, la cual permiten a los empleados utilizar sus dispositivos personales para llevar a cabo tareas de la organización, permitiendo la conexión de estos dispositivos a las redes, servicios o plataformas de dicha organización. Sin embargo, se deben considerar los posibles riesgos que estos pueden traer consigo, debido a que muchos de estos dispositivos no son actualizados de manera oportuna.

El desarrollo de este TFM tiene como objetivo identificar las vulnerabilidades que presentan los dispositivos móviles, a partir de la versión de android y la última actualización de seguridad instalada en el dispositivo, de igual manera se podrán establecer las actualizaciones de seguridad que corrigen dichas vulnerabilidades para luego validar la eficacia de estas permitiendo generar una Whitelist de dispositivos seguros.

Palabras Clave: Android, Bluetooth, Vulnerabilidades, Actualización de seguridad, BYOD.

2 Abstract

Mobile devices are now considered an indispensable tool for the development of personal, school and work activities. Therefore, many organizations are implementing policies like BYOD, which enable employees to use their personal devices to perform company tasks, allowing these devices to be connected to the organization's networks, services, or platforms. Nevertheless, it must be considered the potential risks that these may bring, since many of these gadgets are not updated in a timely manner.

The development of this TFM aims to identify the vulnerabilities that mobile devices present, starting with the android version and the latest security update installed on the device. Likewise, the security updates that correct these vulnerabilities may be implemented to then validate the effectiveness of these allowing the generation of a Whitelist of safe devices.

Keywords: Android, Bluetooth, Vulnerability, Security Update, BYOD

3 Introducción

3.1 Motivación

La seguridad de la información es fundamental en todas las tecnologías de comunicación y La tecnología Bluetooth no es la excepción. Debido al creciente usos de la tecnología inalámbrica, se han descubierto nuevas amenazas (vulnerabilidades) en los dispositivos Bluetooth. Las cuales deben entenderse correctamente para ser abordadas, debido a que estos se utilizan para transferir una amplia gama de información, incluidos datos, vídeos, imágenes, audio y archivos, facilitando algunas actividades de nuestra vida diaria. Sin embargo, por muy ventajosa que sea la tecnología bluetooth para la humanidad en el proceso de emparejar dos dispositivos y establecer una conexión segura, es propensa a una variedad de ataques que pueden afectar no sólo a los dispositivos Bluetooth, sino también a los usuarios. (Albahar, 2017, pp. 17-18)

Dado que, bluetooth a lo largo de su desarrollo e integración con otras tecnologías ha presentado una serie de vulnerabilidades las cuales pueden comprometer la integridad, confidencialidad de los datos y disponibilidad de los servicios que utilizan este estándar de comunicación inalámbrica. Sin embargo, también ha presentado una serie de mejoras y corrección de fallos y vulnerabilidades por parte de los fabricantes que integran esta tecnología en sus dispositivos.

3.2 Planteamiento del problema

Por consiguiente, se pretenden verificar las actualizaciones y parches de seguridad que reciben los dispositivos móviles Android para solventar las vulnerabilidades críticas que presenta la tecnología bluetooth. Las cuales son reportadas en la Base de Datos de Vulnerabilidad Nacional NVD y categorizadas con el estándar V2 y V3.x del Sistema de Puntaje de Vulnerabilidad Común CVSS. Con la finalidad de demostrar el compromiso de la industria en termino de solucionar estos problemas de seguridad.

Todo esto se llevará a cabo en un entorno controlado, iniciando con la clasificación de las vulnerabilidades críticas que presenta la tecnología bluetooth especialmente en la plataforma Android y la clasificación de los diferentes dispositivos que son vulnerables, priorizando los más utilizados en el hogar y las empresas.

Una vez finalizado el proceso de clasificación, se procederá a validar la efectividad de las actualizaciones que recibieron los dispositivos por parte de sus desarrolladores, verificando si fueron corregidas o si aún persisten dichas vulnerabilidades.

Además, de brindarles a los usuarios una whitelist de dispositivos Android con baja probabilidad de explotación de vulnerabilidades a través de la tecnología bluetooth, sirviendo como referente para la escogencia de dispositivos seguros que puedan ser utilizados en la modalidad de trabajo Bring Your Own Devices (BYOD).

3.3 Estructura del trabajo

Este documento consta de la siguiente estructura en capítulos:

3.3.1 Capítulo 2. Estado del arte

Se señalan los conceptos básicos, contexto actual de la tecnología bluetooth y las vulnerabilidades que está presenta, basado en textos e investigaciones de otros autores, de manera que se pueda realizar una clasificación de las vulnerabilidades críticas, identificar algunas técnicas para la explotación de estas y metodologías que permitan validar la efectividad de los parches de seguridad diseñados para esta tecnología en la plataforma Android.

3.3.2 Capítulo 3. Objetivos y metodología de trabajo

Se indica el objetivo general y específicos de éste TFM, así mismo se describen los procesos empleados en la realización del piloto experimental.

3.3.3 Capítulo 4. Desarrollo específico de la contribución

Se describe de forma minuciosa la estructura del laboratorio utilizado para identificar las vulnerabilidades bluetooth que afectan a los dispositivos Android. De igual manera de detallará la metodología utilizada en la explotación de las vulnerabilidades que presenten los dispositivos móviles, permitiendo validar la eficacia de las actualizaciones y parches de seguridad que proveen sus desarrolladores con el fin de corregir las vulnerabilidades de seguridad detectadas. Todo esto se resume en la correlación de los resultados obtenidos y la creación de una Whitelist de dispositivos seguros.

3.3.4 Capítulo 5. Conclusiones y trabajos futuros

Se relatan las conclusiones obtenidas en el desarrollo del TFM y se plantean futuras líneas de investigación para abordar este tipo de problemática.

3.3.5 Capítulo 6. Referencias

Se enumeran las referencias utilizadas en la elaboración del TFM.

3.3.6 Capítulo 7. Anexos

Se adjunta evidencia relevante de las etapas y procesos realizados en forma de capturas de pantalla, ficheros imagen etc.

4 Estado del arte

4.1 Marco teórico

4.1.1 Descripción De La Tecnología Bluetooth

Cuando hablamos de tecnología bluetooth, hacemos referencia a una tecnología de transmisión inalámbrica por medio de ondas de radio de corto alcance (1, 20 y 100 m a la redonda dependiendo la versión), con capacidad para la transmisión tanto de datos como de voz a más de 720 Kbps por canal (Sparacino, 2003)

El principal objetivo de esta tecnología es la posibilidad de reemplazar los muchos cables propietarios que conectan unos dispositivos con otros por medio de un enlace radio universal de corto alcance. Esta tecnología, es única en cuanto a: la cantidad de aplicaciones que puede tener, la forma de los enlaces (simultaneo, por grupo de productos, o entre productos individuales a internet); lo que ha permitido combinada con lo estrictos requerimientos de interoperabilidad ha permitido que diferentes segmentos comerciales sirvan de soporte para la tecnología bluetooth, incluyendo fabricantes de software, cámara, computadoras móviles, carros, equipos electrónicos (Vergara, 2008).

La tecnología Bluetooth ofrece un puente a las redes de datos existentes, una interfaz con el exterior y un mecanismo para formar en el momento, pequeños grupos de dispositivos conectados entre sí de forma privada fuera de cualquier estructura fija de red, formado por un pequeño transmisor de radiofrecuencia; dicho transmisor está integrado en un pequeño microchip y opera en una frecuencia de banda global (2,4 GHz) que asegura la compatibilidad universal (Castellano, 2012)

4.1.2 Desarrollo De La Tecnología Bluetooth

En 1994, Ericsson Mobile Communications inicia el desarrollo de la tecnología bluetooth, con el objetivo de crear una interfaz de radio de baja potencia y bajo costo que permitiera la comunicación entre los teléfonos móviles y sus accesorios, con el fin de eliminar los cables entre los teléfonos móviles, auriculares y otros dispositivos. Para que dicho sistema fuera exitoso y verdaderamente utilizable, se debía implementar la tecnología de radio enlaces de corto alcance en una cantidad critica de dispositivos portátiles. Por lo que, en febrero de 1998, cinco compañías, Ericsson, Nokia, IBM, Toshiba e Intel, forman un Grupo de Interés Especial

(SIG, por sus siglas en ingles). Dicho grupo, logro establecer la creación de una especificación global para conectividad sin hilos de corto alcance (Vergara, 2008)

4.1.3 Vulnerabilidad

La definición de vulnerabilidad en términos generales se cataloga como un fallo en un sistema que puede ser explotado por un atacante generando un riesgo para la organización o pudiendo afectar la confidencialidad, integridad o disponibilidad del mismo sistema. Existen dos tipos de vulnerabilidades física y lógica (Romero Castro et al., 2018, p. 41)

4.1.3.1 Vulnerabilidades Física

Este tipo de vulnerabilidades son la que afectan la infraestructura de la organización de manera física. Dentro de estos encontramos los desastres naturales, los cuales son eventos fortuitos que pueden ocasionar la negación o la afectación de la disponibilidad de un servio. Otra de las opciones físicas son los controles de acceso, en muchas ocasiones se tiene los accesos a la infraestructura crítica y no se tiene los accesos pertinentes, cualquier persona podría abrir una puerta, podría entrar y constituye un gran riesgo para la organización porque cualquier usuario podría ingresar con una USB y copiar información confidencial, o en su defecto podría infectar la misma infraestructura.

4.1.3.2 Vulnerabilidades Lógica

Este tipo de vulnerabilidades son las que afectan directamente la infraestructura y el desarrollo de la operación de estos, como pueden ser: las vulnerabilidades de configuración; se presenta cuando se conservan las configuraciones por defecto del sistema o incluso de algunas aplicaciones del servidor que se tenga expuesta, también puede la configuración de algunos firewalls que no está gestionado de manera correcta. Vulnerabilidades de actualización; se presenta cuando una organización o empresa no actualizan sus sistemas, quedando expuesto a nuevas vulnerabilidades. Las vulnerabilidades de desarrollo; aquí se puede mencionar las inyecciones de código en SQL, Cross Site Scripting, esto puede variar dependiendo del tipo de aplicación y validación de los datos.

4.1.4 Base de Datos de Vulnerabilidad Nacional (NVD).

El NVD es un repositorio del gobierno de los estados unidos, que permite la gestión de vulnerabilidades basándose en estándares representados mediante el Protocolo de automatización de contenido de seguridad (SCAP). Estos datos permiten la automatización de la gestión de vulnerabilidades, la medición de la seguridad y el cumplimiento. El NVD

incluye bases de datos de referencias de listas de comprobación de seguridad, defectos de software relacionados con la seguridad, configuraciones incorrectas, nombres de productos y métricas de impacto.

4.1.5 Sistema de puntuación de vulnerabilidad común (CVSS)

CVSS es un sistema de puntuación que proporciona un método estándar y abierto para estimar el impacto de una vulnerabilidad y que se compone tres grupos principales de métricas: Base, Temporal y de Entorno. Cada uno de estos grupos se compone a su vez de un conjunto de métricas las cuales pueden variar de acuerdo con la versión ya sea Cvssv2 o Cvssv3. (INCIBE-CERT, 2015)

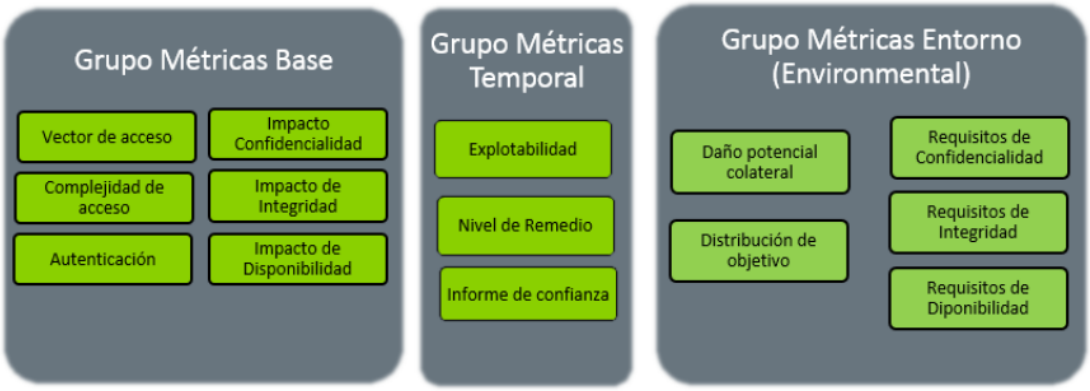

Ilustración 1. Métricas del sistema de puntuación CVSS v2 (Fuente: https://www.incibe-cert.es/sites/default/files/blog/cvss3-0/metricasv2.png)

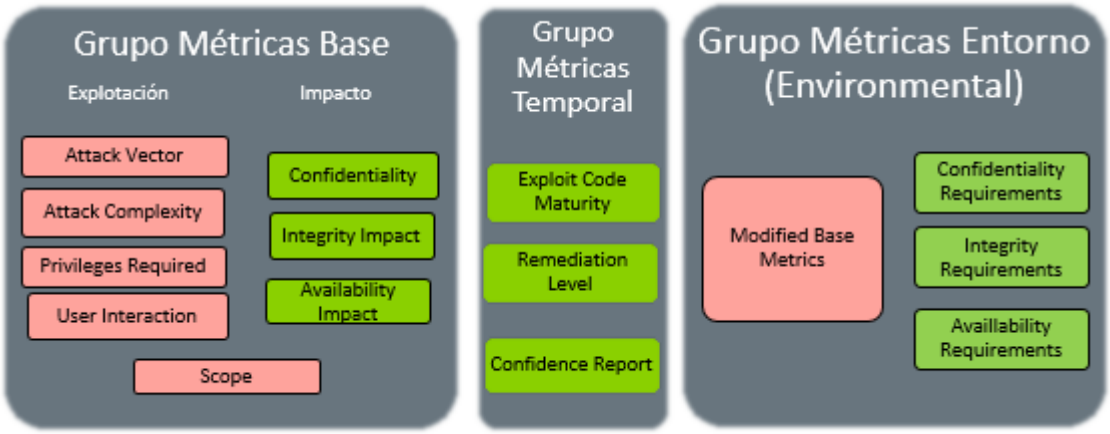

Ilustración 2. Métricas del sistema de puntuación CVSS v3 (Fuente: https://www.incibe-cert.es/sites/default/files/blog/cvss3-0/grupos_metricas_cvss3.png)

4.1.6 Proyecto de Código Abierto Android (AOSP)

Android es un sistema operativo para dispositivos móviles y un proyecto de código abierto dirigido por Google, el cual se encarga de proporciona los repositorios, información y el código fuente necesarios para crear variantes personalizadas del sistema operativo Android, dispositivos de puerto y accesorios para la plataforma Android, garantizando que los dispositivos cumplan con los requisitos de compatibilidad que mantienen el ecosistema de Android un entorno saludable y estable para millones de usuarios (Android Partner Docs, 2019)

4.1.7 Pentesting

Es un ataque simulado y autorizado contra un sistema informático que tiene como objetivo de evaluar la seguridad del sistema. Durante la prueba, se busca identificar las vulnerabilidades presentes en el sistema y se explotarlas tal como haría un atacante con fines maliciosos. Con el fin de evaluar los riegos que pueda presentar el sistema y sugerir un plan de medidas correctivas (Zafra y Luis, 2017)

4.1.8 Tipos de Pentesting

Black-Box o Caja Negra; en este enfoque el pentester no tiene ninguna información sobre el sistema que se desea auditar y tendrá que hacer uso de los recursos que disponga para obtener la mayor información posible de su objetivo e intentar comprometer el mismo. White-Box o Caja Blanca; en este enfoque el pentester recibe gran cantidad de información del sistema que va a auditar como puede ser la topología de red, rangos de IP, sistemas operativos. Permitiendo ahorrar tiempo en la fase de recolección de información lo que facilita de cierta manera la tarea de intrusión de tal forma que este pueda ser capaz de encontrar vulnerabilidades en el sistema. Por último, Grey-Box o Caja Gris; este trata de combinar los dos primero enfoques ofreciendo únicamente información meramente orientativa al pentester (Zafra y Luis, 2017)

4.1.9 Singer Board Computer (SBC)

Una computadora de placa única o SBC (Single Board Computer), por sus siglas en inglés, es una computadora la cual se construye sobre una única placa que contiene componentes electrónicos tales como microprocesadores, memoria, dispositivos de entrada/salida y cualquier otro componente requerido para ser un computador completamente funcional. Estas son comúnmente utilizadas en demostraciones o prototipos, como herramienta con fines

educacionales, o como un sistema embebido, funcionando como unidades que proveen control y sirven de interfaz en sistemas más complejos como cadenas de producción, robótica, redes de datos y seguridad informática (Moreno y Ramírez, 2017)

Una de las placas SBC que ha tomado auge en los últimos años por su bajo costo y alto rendimiento es la placa Raspberry Pi, la cual cuenta con una variedad de modelos tales como Raspberry pi 1, Raspberry pi 2, Raspberry pi 3, Raspberry pi 4, entre otros. Diseñada por la Fundación Raspberry Pi. Organización benéfica con sede en el Reino Unido que trabaja para poner el poder de la informática y la creación digital en manos de personas de todo el mundo en aras de mejorar la educación y el desarrollo de herramientas tecnológicas.(*Raspberry Pi Foundation*, s. f.)

4.1.10 Kali Linux

Kali Linux se lanzó el 13 de marzo de 2013 como una reconstrucción completa de arriba a abajo de BackTrack Linux, que se adhiere completamente a los estándares de desarrollo de Debian. Destinada para llevar a cabo auditorías de seguridad, ya que cuenta con varios cientos de herramientas que están orientadas a diversas tareas de seguridad de la información, como pruebas de penetración, investigación de seguridad, informática forense e ingeniería inversa. este proyecto es desarrollado, financiado y mantenido por Offensive Security, una compañía líder en capacitación en seguridad de la información (g0tmi1k, 2019)

4.1.11 Bring Your Own Devices (BYOD)

BYOD es una estrategia propuesta por el director de seguridad y privacidad de Intel Malcolm Harkins en 2009. Este concepto permite a los empleados utilizar tecnología de propiedad personal para realizar tareas propias de su Organización. De tal manera que estos puedan acceder a los servicios y/o datos proporcionados por el empleador sobre sus tabletas personales, e-Readers, teléfonos inteligentes u otros dispositivos. Uno de los argumentos a favor de BYOD es que es un utilizar eficazmente los recursos. Ya que los computadores de las oficinas permanecen sin utilizarse cuando el horario de oficina ha terminado (Afreen, 2014)

4.2 Antecedentes

La primera versión de dicha tecnología (1.0) se publicó en julio de 1999, desde ahí hasta la fecha, más de 1300 fabricantes a nivel mundial de diferentes áreas de negocios se han unido a la familia bluetooth, posibilitando que dicha tecnología sea el prototipo industrial de mayor y más rápido crecimiento (Outeiriño et al., 2004), cuyos principales avances se detallan a continuación:

4.2.1 Bluetooth V1

Bluetooth alcanzó su primera versión en 2002 bajo el nombre de IEEE 802.15.1 o bluetooth V1.1 ya que las versiones 1 y 1b no fueron consideradas debido a la gran cantidad de problemas de compatibilidad que presentaron. Dicha versión además de resolver los problemas de compatibilidad agrego nuevas especificaciones como es el soporte para un rango de conexiones 10/100, permitiendo el uso de la tecnología de transmisión de datos robusta, mediante espectro ensanchado de saltos de frecuencia (FHSS); que soportaba transmisión simultánea de voz y datos con aproximadamente 1 Mbit / s (720 kbps) de velocidad de datos (Bisdikian, 2001)

Por otra parte, se definen tres topologías como son; punto a punto, de una sola célula (piconet) y multi celda (scatternet); las cuales permiten la conexión de hasta ocho dispositivos de datos activos en una piconet, donde cada piconet tiene un maestro y el resto deben servir como esclavos quienes solo tienen vínculos con el maestro, algo importante de denotar es que hasta 10 piconets pueden existir dentro de un rango de 10 metros, además estos pueden estar de vez en cuando en modo "stand by" (no activo). Dentro de las limitaciones de esta versión podemos encontrar que no se ocupa del enrutamiento, la mayoría de las funciones de red se empujan a la capa de enlace, no es compatible con multi-hop multicasting, el definido nodo maestro es un cuello de botella, el número de nodos activos en una piconet se limita y no se preocupa por ahorro de energía ya hecho en las capas superiores (Muller, 2002)

4.2.2 Bluetooth V2.0 + EDR

Versión lanzada en 2004, la cual es compatible con la versión anterior V1.2. Su principal diferencia es la introducción de una velocidad de datos mejorada EDR (Enhanced Data Rate, en inglés), para acelerar la transferencia de datos y puede proporcionar un menor consumo de energía a través de un ciclo de trabajo reducido. Esta versión, incluyen mejores sistemas para la transmisión de datos, además, posee las características funcionales de la v1.2, y agrega dos nuevos sistemas de modulación. Los paquetes EDR proporcionan velocidades

máximas de transmisión de datos de 2 y 3 Mbps; El incremento en la velocidad se consigue gracias al (DPSK) aumenta tres veces los bits por la cantidad de símbolos trasmitidos. También proporcionan trasmisión simultánea en la misma piconet y los dispositivos emplean el mismo código de acceso, cabecera y sistema de saltos de frecuencia (Avila y Reyes, 2017).

4.2.3 Bluetooth V2.1 + EDR

Al igual que la versión V2.0 esta sigue siendo totalmente compatible con la V1.2, esta versión fue adoptada por el Bluetooth SIG (Bluetooth Special Interest Group) el 26 de julio de 2007. La función principal para destacar en esta versión es el emparejamiento simple y seguro SSP (Secure Simple Pairing, en inglés), que mejora la experiencia de emparejamiento entre los dispositivos Bluetooth, aumentado el uso y seguridad de este. Además incluye EIR, que proporciona más información durante el procedimiento de investigación para permitir un mejor filtrado de los dispositivos antes de la conexión y reduciendo el consumo de energía en modo de bajo consumo (Avila y Reyes, 2017).

4.2.4 Bluetooth V3.0

La versión 3.0 soporta velocidades de transferencia de datos de 24 Mbit. Entre sus principales características es que cuando deben enviarse grandes cantidades de datos se utiliza PHY MAC 802.11 Que generalmente están asociados con la tecnología wifi para transportar los datos. El núcleo del sistema bluetooth consiste en un host y uno o más controladores, en el que el host son todas las capas por debajo de los perfiles y por encima de la interfaz del controlador HCI (Host Controller Interface) (Morales, 2011).

4.2.5 Bluetooth V4.0

La versión 4 o BLE por sus siglas en ingles de “Bluetooth Low Energy”, funciona muy parecido al bluetooth convencional en cuanto a los procedimientos, utilizando los mismos procedimientos de “anuncio” y “sincronización”; además, usa la misma técnica del salto de frecuencia de espectro ensanchado FHSS y opera bajo la misma banda frecuencias ISM que el Bluetooth clásico (2.4 a 2.48 Ghz). “Pero este utiliza una distribución distinta de canales: en Bluetooth clásico se utilizan 79 canales de 1 Mhz de ancho de banda mientras que en BLE se utilizan 40 de 2 MHz de ancho de banda” (Avila, 2016).

Algo importante en las diferencias en funcionamiento con el bluetooth clásico es que BLE en arquitectura no se permite las “redes de dispersión” o “scatternet” del Bluetooth tradicional, “siendo en topología de estrella el único modo de creación de redes. Aun así́, un solo “máster”

podría tener prácticamente ilimitados esclavos (la asignación de direcciones se realiza con 48 bits, lo que supone alrededor de 218 billones de direcciones distintas)(Morales Pedro, 2011)

Pero la diferencia que más denota del BLE con el bluetooth tradicional es que el consumo de energía ("la especificación dice que BLE puede llegar a necesitar solo el 10% de lo que necesita Bluetooth para Funcionar"). Lo que hace posible tan bajo consumo de energía, con respecto a la del Bluetooth clásico, es que BLE es un protocolo "sin conexión", es decir, los dos extremos de la comunicación no tienen la necesidad de estar permanentemente conectados. Quiere decir que esto "permite a los dispositivos permanecer "dormidos" o en estado de "standby" durante mucho tiempo en caso de que no haya información que intercambiar, lo que reduce significativamente la cantidad de tiempo que el dispositivo y su parte transmisora/receptora necesita estar encendida" (Akhayad, 2016)

4.2.6 Bluetooth Versión 4.1

Bluetooth 4.1 se lanzó en 2013. En esta se incorpora un cambio fundamental con respecto al soporte de red de malla BLE; un dispositivo, independientemente de su función de capa de enlace, puede ejecutar varias instancias de capa de enlace simultáneamente sin limitación. Por lo tanto, se permite que un esclavo se conecte simultáneamente a más de un maestro. Además, un dispositivo puede actuar como esclavo en ciertos intervalos y como maestro en otros, manteniendo comunicaciones paralelas con sus vecinos. (Darroudi y Gomez, 2017)

4.2.7 Bluetooth Versión 4.2

Se publicó en 2014, incorporando mejoras en tres áreas; Conectividad a Internet, seguridad mejorada y mayor rendimiento. Estas actualizaciones están destinadas a aumentar las posibilidades de BLE como tecnología para IoT. Sin embargo, Bluetooth 4.2 no proporciona más funcionalidad para admitir redes de malla BLE.(Darroudi y Gomez, 2017)

4.3 Estado actual

4.3.1 Bluetooth V5.0

Es publicado a finales de 2016, siendo la última especificación de Bluetooth lanzada. La cual ofrece mejoras en términos de alcance, velocidad de datos y funcionalidad del canal publicitario. Este último comprende un aumento de la capacidad de mensajes publicitarios, junto con la definición de dos tipos de canales publicitarios, primario y secundario. Los canales

publicitarios primarios son los mismos tres canales publicitarios disponibles en versiones BLE anteriores, mientras que los canales publicitarios secundarios usan los 37 canales BLE restantes (anteriormente definidos únicamente como canales de datos). Sin embargo, al igual que Bluetooth 4.2, no proporciona más funcionalidad para admitir redes de malla BLE más allá de las de Bluetooth 4.1(Darroudi y Gomez, 2017)

4.3.2 Vulnerabilidades bluetooth

La seguridad de Bluetooth es actualmente un área de investigación muy activa tanto en la academia como en la industria. Al igual que cualquier otro sistema de comunicación inalámbrico, la transmisión Bluetooth puede ser interceptada o bloqueada deliberadamente. Refiriéndose a las vulnerabilidades, aunque la tecnología Bluetooth tiene un cierto grado de seguridad, tiene un número considerable de fallas, estas vulnerabilidades generalmente se deben a fallas predeterminadas, como la fuerza del generador de números aleatorios y el uso de un valor de PIN pequeño. Otros factores que contribuyen a la aparición de vulnerabilidades son las fallas de implementación, así como la flexibilidad del estándar que permite a los fabricantes autonomía para definir varios procedimientos relacionados con la criptografía y la autenticación (Rocha y Bruno, 2006). Así, las vulnerabilidades en el sistema Bluetooth se pueden dividir en tres categorías principales de la siguiente manera (Mina y Tarique, 2012)

- Amenaza de divulgación: la información puede filtrarse del sistema de destino a un espía que no está autorizado para acceder a la información.
- Amenaza de integridad: la información puede modificarse deliberadamente para inducir a error al destinatario.
- Amenaza de negación de servicio (DoS): los usuarios pueden ser bloqueados para obtener acceso a un servicio, haciéndolo no disponible o limitando severamente su disponibilidad a un usuario autorizado.

4.4 Trabajos relacionados

4.4.1 BlueBorne

Se denomino BlueBorne ha ochos vulnerabilidades zero-day identificadas por Armis Labs, que indican la existencia y el potencial de este vector de ataque mediante el cual los piratas informáticos pueden aprovechar las conexiones Bluetooth para penetrar y tomar el control completo sobre los dispositivos objetivo. Ya que afecta computadores comunes, los teléfonos móviles y el ámbito en expansión de los dispositivos IoT. El ataque no requiere que el

dispositivo objetivo esté emparejado con el dispositivo del atacante, o incluso que esté configurado en modo detectable. Estas vulnerabilidades son completamente operativas y pueden explotarse con éxito y se puede usar para llevar a cabo una amplia gama de delitos, incluida la ejecución remota de código, así como los ataques Man-in-The-Middle (Ben Seri y Gregory Vishneplsky, 2017)

4.4.2 Bluetooth Vulnerability Finder:

Este escáner de vulnerabilidades en la tecnología BTL IoT (Bluetooth Low Energy Internet of Things), puede diagnosticar el estado de seguridad de los dispositivos IoT sin necesidad de usar algún hardware especializado. La aplicación analiza los anuncios BLE de los dispositivos e informa sobre su nivel de seguridad clasificando el dispositivo en uno de cuatro niveles e ilustra la criticidad a través de un código de colores: Rojo - Crítico; Naranja - Alto; Amarillo - Medio; Verde - Bajo. (Betancourt Duque y Peña Salazar, 2019)

5 Objetivos y Metodología del trabajo

5.1 Objetivo general

Verificar la efectividad de las actualizaciones de seguridad de la tecnología bluetooth en los dispositivos Android teniendo en cuenta algunas de las vulnerabilidades reportadas en la NVD y clasificadas como crítica con el CVSS versión (V3.x) y/o alta con el CVSS versión (V2), para obtener Whitelist de dispositivos seguros que puedan ser utilizados en la modalidad de trabajo (Bring Your Own Devices).

5.2 Objetivos específicos

- Determinar las vulnerabilidades críticas que presenta la tecnología bluetooth en la plataforma Android.
- Explotar las vulnerabilidades críticas que presenta la tecnología bluetooth en la plataforma Android después de las actualizaciones de seguridad.
- Correlacionar las vulnerabilidades que presenta la tecnología bluetooth en la plataforma Android antes y después de las actualizaciones.
- Construir una Whitelist de dispositivos seguros que puedan ser utilizados en la modalidad de trabajo (Bring Your Own Devices).

5.3 Metodología

Para el desarrollo del siguiente Trabajo de Fin de Máster (TFM) se ha decidido aplicar una metodología piloto experimental la cual se dividió en 4 fases tales como:

1. Revisión bibliográfica de vulnerabilidades bluetooth en dispositivos móviles android.
2. Identificación de las vulnerabilidades Bluetooth en los dispositivos Android.
3. Actualización de los dispositivos vulnerables y validación de las actualizaciones.
4. Informe de resultados y creación de Whitelist de dispositivos seguros.

Inicialmente se realizará una búsqueda de información de manera sistémica que permita entender el funcionamiento de la tecnología bluetooth y como es afectada la seguridad

de los dispositivos Android que poseen esta tecnología. De igual manera se determinarán las herramientas y técnicas empleadas en la identificación y explotación de vulnerabilidades en dispositivos móviles.

Seguidamente se establecerán una metodología para la identificación, categorización y explotación de las vulnerabilidades bluetooth detectadas en dispositivos móviles mediante el uso de la aplicación BlueBorne, la cual mediante un escaneo permite clasificar el nivel de vulnerabilidad que presenta un dispositivo con tecnología bluetooth. Asimismo, se utilizarán cualesquiera de las técnicas (Blueprinting, BT Audit, Snarfing, Man in The Middle), acompañado de script y algunas de las herramientas que no ofrece Kali Linux. De manera que, se pueda validar la efectividad de las actualizaciones instalada en cada uno de los dispositivos móviles analizados.

Por último, se realizará la correlación de los resultados obtenidos antes y después de aplicar la metodología de explotación de vulnerabilidades, con la finalidad de establecer una Whitelist de dispositivos seguros que puedan ser utilizados en el modelo de trabajo (Bring Your Own Devices).

6 Desarrollo especifico de la contribución

6.1 Descripción del piloto

El desarrollo del siguiente piloto se llevará a cabo en un entorno controlado que cuenta con los siguientes dispositivos. Un computador de escritorio con el sistema operativo Windows 10, en el cual se instalarán programas tales como: **SD Cart Formatter**, empleado para darle formato a la memoria Ssd y así poder utilizar todo el espacio de está, eliminando cualquier partición encontrada; **Win32diskimager,** utilizado para copiar imágenes de arranque en dispositivos flash USB o SD.

También se utilizará una Raspberry PI 4 que funcionará como herramienta portable para el análisis y explotación de las vulnerabilidades en los dispositivos móviles, donde montaremos el sistema operativo ***Kali Linux,*** distribución basada en Debian GNU/Linux. Que posee un conjunto de programas necesarios para llevar a cabo la explotación de las vulnerabilidades detectadas en los dispositivos móviles que se utilizarán en el piloto, teniendo en cuenta la cuota de mercado global de teléfonos inteligentes según la investigación realizada en el tercer trimestre de 2019 por la consultora Strategy Analytics (Yiwen Wu, 2019).

Tabla 1
Posicionamiento Global de Marcas de celulares Inteligentes 3Q 2019.

Marca	Porcentaje
Samsung	21.3 %
Huawei	18.2 %
Apple	12.4 %
Xiaomi	8.8 %
OPPO	8.0 %
Otros	31.1 %

Recuperado del informe Global Smartphone Shipments Return to 2 Percent Growth in Q3 2019 que presento la firma Strategy Analytics (Fuente: Strategy Analytics, 2019)

Tabla 2
Equipos de Cómputo Empleados en el Piloto.

Nombre	Marca	Modelo	Sistema Operativo
Equipo_1	HP	ProDesk 400G5 SFF	Windows 10
SBC_1	Raspberry Pi 4	Modelo B 4GB RAM	Kali Linux

Descripción detallada de los equipos empleados para el desarrollo del piloto, (Fuente: Elaboración propia).

Tabla 3
Dispositivos Móviles Empleados en el Piloto.

Dispositivo	Marca	Modelo	Versión de Android
Dispositivo 1	LG	LG-K430	Android 6.0
Dispositivo 2	Motorola	Moto G[5]	Android 7.0

Se construye a partir de los datos de la tabla 1. Tomando las dos primeras marcas de celulares inteligentes y otra marca adicional como referencia para el desarrollo del piloto. Además, se realiza la clasificación de los dispositivos móviles por modelo y versión de android, (Fuente: elaboración propia).

6.2 Preparación del ambiente base

Para iniciar con el desarrollo de este piloto, se implementará una herramienta portable que permitirá realizar el análisis de vulnerabilidades en los dispositivos móviles. Para esto inicialmente se construirá una estructura plástica utilizando impresión 3D la cual permitirá acoplar la SBC Raspberry pi 4 Modelo B con una pantalla Lcd touch screen de 3.2 pulgadas.

Ilustración 3. Impresión 3D Caja Raspberry Pi con Adaptación de Pantalla Lcd, (Fuente: Elaboración Propia).

Seguidamente pasamos a la instalación, configuración y actualización del sistema operativo Kali Linux sobre la SBC Raspberry pi 4 Modelo B, ver Anexo 1.

Ilustración 4. Dispositivo Portable Para la Identificación y Explotación de Vulnerabilidades Bluetooth, (Fuente: Elaboración Propia).

6.3 Desarrollo del piloto

Una vez configurado el dispositivo portable continuamos con la aplicación de la metodología propuesta para el desarrollo del piloto.

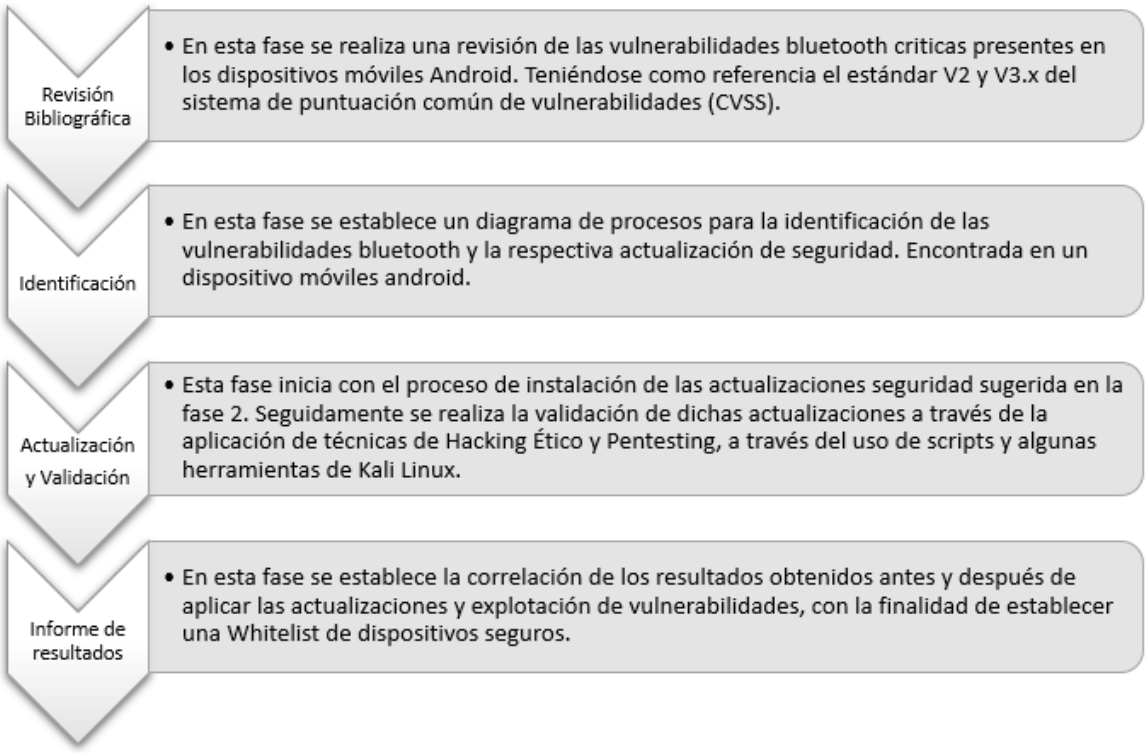

Ilustración 5. Metodología Para la Identificación de Vulnerabilidades Bluetooth y Validación de los Patch de Seguridad, Dispuesto Para los Dispositivos Móviles Android. (Fuente: Elaboración Propia.)

6.3.1 Clasificación de vulnerabilidades bluetooth y Patch de seguridad.

Este proceso inicia con una revisión de vulnerabilidad en la Base de Datos de Vulnerabilidad Nacional (NVD). En la cual se realiza una consulta teniendo en cuenta la configuración de algunos parámetros de la búsqueda avanzada, arrojando como resultado un total de 64 vulnerabilidades que afectan a la tecnología bluetooth con puntajes crítico y alto según el CVSS (V3.x y V2), de las cuales solo 16 afectan a los dispositivos android.

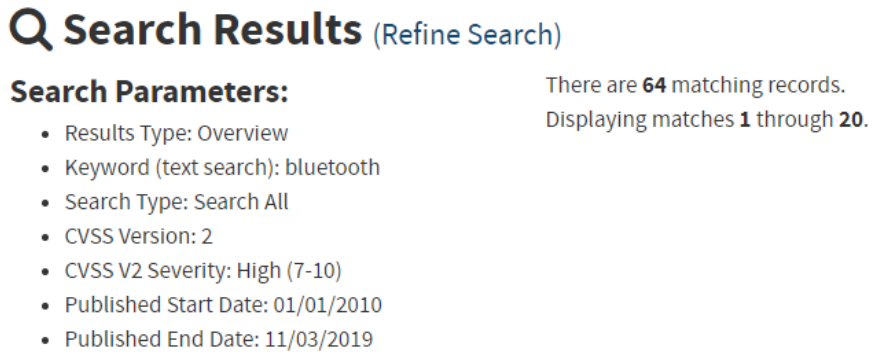

Ilustración 6. Resultado de la Búsqueda de Vulnerabilidades Bluetooth con Puntuación de Gravedad Alta. (Fuente: Elaboración Propia).

Tabla 4
Vulnerabilidades Bluetooth en Dispositivos Android

CVE	Versión de Android	CVSS V2	CVSS V3
CVE-2019-9365	Android 10	7,5	9,8
CVE-2019-9259	Android 10	7,2	6,7
CVE-2019-2009	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	8,3	8,8
CVE-2019-2102	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	8,3	8,8
CVE-2019-2049	Android 9	7,2	7,8
CVE-2018-9583	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	10	9,8
CVE-2018-9555	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	8,3	8,8
CVE-2018-9363	Android kernel ID de Android: A-65853588	7,2	8,4
CVE-2018-9358	Android 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0, y 8.1	7,8	7,5
CVE-2018-9504	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	8,3	8,8
CVE-2018-9476	Android 8.0 y 8.1	10	9,8
CVE-2017-13283	Android 7.0, 7.1.1, 7.1.2, 8.0 y 8.1	10	9,8

CVE-2017-13160	Android 7.0, 7.1.1, 7.1.2 y 8.0	10	9,8
CVE-2017-0842	Android 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2 y 8.0	7,2	7,8
CVE-2017-0782	Android 4.4.4, 5.0.2, 5.1.1, 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2 y 8.0	8,3	8,8
CVE-2017-0781	Android 4.4.4, 5.0.2, 5.1.1, 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2 y 8.0	8,3	8,8

Construida a partir de una consulta de vulnerabilidades en la tecnología bluetooth, realizada en la Base de Datos de Vulnerabilidad Nacional NVD https://nvd.nist.gov/vuln/search/ (Fuente: elaboración propia)

Tabla 5
Vulnerabilidades Bluetooth con Puntaje Crítico y Alto que Afectan Dispositivos Android.

CVE	**Versión de Android**	**CVSS V2**	**CVSS V3.x**
CVE-2019-2009	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	8,3	8,8
CVE-2019-2102	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	8,3	8,8
CVE-2018-9583	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	10	9,8
CVE-2018-9555	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	8,3	8,8
CVE-2018-9504	Android 7.0, 7.1.1, 7.1.2, 8.0, 8.1 y 9	8,3	8,8
CVE-2018-9476	Android 8.0 y 8.1	10	9,8
CVE-2017-13283	Android 7.0, 7.1.1, 7.1.2, 8.0 y 8.1	10	9,8
CVE-2017-13160	Android 7.0, 7.1.1, 7.1.2 y 8.0	10	9,8
CVE-2017-0782	Android 4.4.4, 5.0.2, 5.1.1, 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2 y 8.0	8,3	8,8
CVE-2017-0781	Android 4.4.4, 5.0.2, 5.1.1, 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2 y 8.0	8,3	8,8

Construida a partir de los datos presentados en la tabla 3, permitiendo la clasificación de vulnerabilidades criticas teniendo en cuenta el rango de gravedad definido en las especificaciones del CVSS v3.x y otras altas que afectan gran parte de las versiones de android, las cuales una vez explotadas pueden llegar a ser consideradas críticas debido al impacto negativo que pueden ocasionar en una organización, (Fuente: autoría propia).

Una vez identificadas las vulnerabilidades, se procede a realizar una revisión de los parches de seguridad emitidos por Google en aras de mejorar el Proyecto de Código Abierto Android (AOSP), presentado soluciones para las vulnerabilidades reportadas en NVD. Los cuales se clasificarán y categorizarán, teniendo en cuenta las versiones de android y las empresas asociadas que se muestra en la tabla 1.

Tabla 6.
Fecha de lanzamiento de los Patch de seguridad emitidos por AOSP y algunos fabricantes de dispositivos móviles android a nivel global.

CVE	Vers. Patch de Seguridad (Google)	Vers. Android Actualizadas	Vers. Patch de Seguridad (Fabricante)
CVE-2019-2009	2019-03-01	7.0, 7.1.1, 7.1.2, 8.0, 8.1, 9	Samsung, SMR-MAR-19 Huawei, EMUI- MAR-19 LG, SMR- MAR-19
CVE-2019-2102	2019-06-01	7.0, 7.1.1, 7.1.2, 8.0, 8.1, 9	Samsung, SMR- JUN-19 Huawei, EMUI- JUN-19 LG, SMR- JUN-19
CVE-2018-9583	2019-01-01	7.0, 7.1.1, 7.1.2, 8.0, 8.1, 9	Samsung, SMR-JAN-19 Huawei, EMUI- JAN-19 LG, SMR- JAN-19
CVE-2018-9555	2018-12-01	7.0, 7.1.1, 7.1.2, 8.0, 8.1, 9	Samsung, SMR- DEC-18 Huawei, EMUI- DEC-18 LG, SMR- DEC-18
CVE-2018-9504	2018-10-01	7.0, 7.1.1, 7.1.2, 8.0, 8.1, 9	Samsung, SMR- OCT-18
CVE-2018-9476	2018-10-01	8.0, 8.1	Huawei, EMUI-OCT-18 LG, SMR- OCT-18
CVE-2017-13283	2018-04-01	7.0, 7.1.1, 7.1.2, 8.0, 8.1	Samsung, SMR-APR-2018 Huawei, EMUI- APR-2018 LG, SMR-APR-2018
CVE-2017-13160	2017-12-01	7.0, 7.1.1, 7.1.2, 8.0	Samsung, SMR- DEC-2017 Huawei, EMUI-DEC-17 LG, SMR-DEC-2017
CVE-2017-0782 CVE-2017-0781	2017-09-05	4.4.4, 5.0.2, 5.1.1, 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0	Samsung, SMR-SEP-2017 LG, SMR-SEP-2017

Construida a partir de los boletines de seguridad publicados por AOSP, Samsung, Huawei, y LG en su página web. Donde se evidencian las fechas en las cuales se realizaron los lanzamientos de las actualizaciones de seguridad que permiten mitigar las vulnerabilidades críticas de la tecnología bluetooth en dispositivos android, (Fuente: elaboración propia).

Después de realizar una revisión bibliográfica, clasificación de vulnerabilidades críticas y actualizaciones de seguridad, pasamos al proceso de identificación de vulnerabilidades, validación de las actualizaciones de seguridad y creación de whitelist de dispositivos móviles seguros. Para lo cual se utilizará el siguiente mapa de procesos, ver ilustración 7.

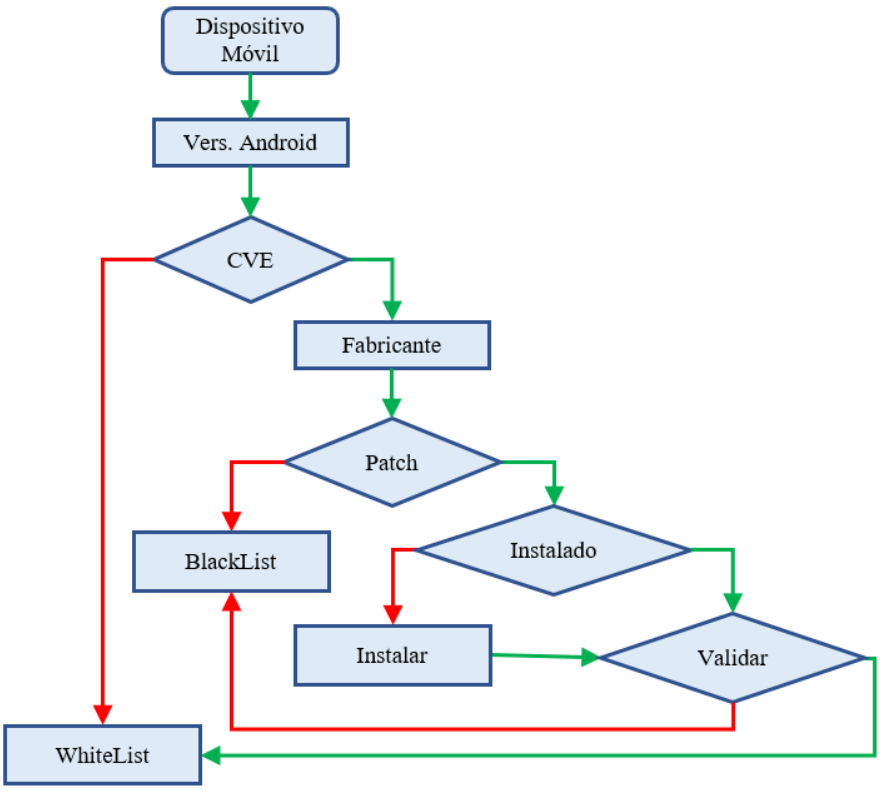

Ilustración 7. Proceso Para la Identificación de Vulnerabilidades, Validación de Actualizaciones de Seguridad en Dispositivos Móviles y creación de Whitelist y Blacklist. (Fuente: Elaboración Propia)

6.3.2 Identificación de vulnerabilidades en dispositivos móviles android.

Para llevara a cabo este proceso se ha diseñado un aplicativo web (BlueExploitApp), que cuenta con una serie de script escritos en el lenguaje de programación Python que automatizan los procesos de identificación de vulnerabilidades e instalación de parches de seguridad, facilitando el análisis y administración de los dispositivos móviles android empleados en una empresa que apliqué la política Bring Your Own Devices (BYOD). Además, permite llevar un registro de los dispositivos analizados, permitiendo generar reportes como:

- Dispositivos que cuentan con la última actualización de seguridad.
- Dispositivos sin actualizaciones de seguridad
- Actualizaciones disponibles por dispositivo.
- Vulnerabilidades que pueden afectar la seguridad del dispositivo.

Siguiendo con el análisis de las vulnerabilidades se procede con la instalación y ejecución del aplicativo BlueExploitApp ver anexo 2.

Una vez ejecutada la aplicación, se realiza la conexión de los dispositivos con el dispositivo portable a través de un cable de datos, es necesario que el dispositivo móvil tenga habilitada la opción que permite la depuración USB, ver anexo 3.

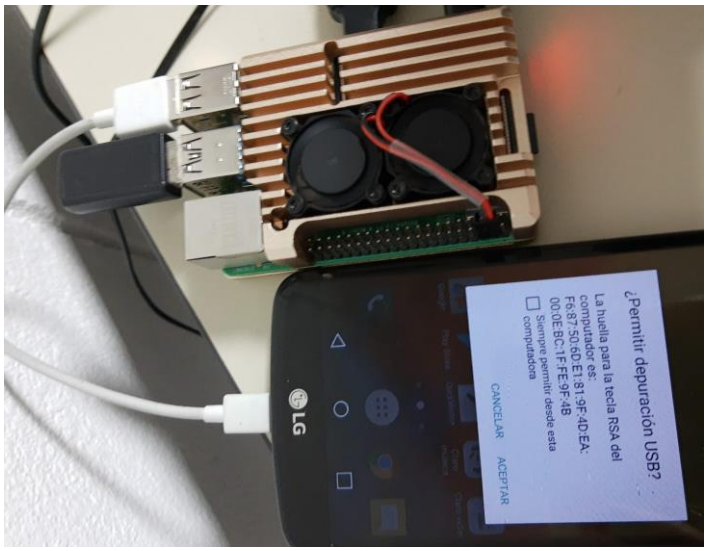

Ilustración 8. Conexión del Dispositivo 1 (LG-K430) y el Dispositivo Portable (Raspberry Pi 4) (Fuente: Elaboración Propia).

Seguidamente, se inicia el proceso de análisis de los dispositivos con la aplicación BlueExploitApp, se escoge la opción Dispositivo del menú principal, se presiona el botón Escanear Dispositivo, para obtener el listado de dispositivos conectados y por último presionamos el botón seleccionar el cual ejecutara un script que capturara los datos necesarios para luego iniciar la identificación de vulnerabilidades.

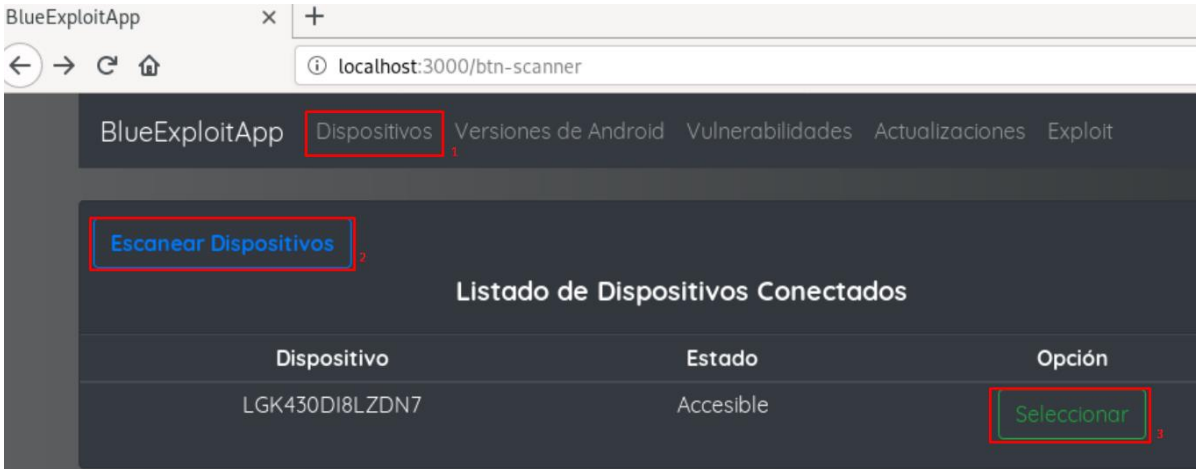

Ilustración 9. Dispositivos Conectados con la Aplicación BlueExploitApp (Fuente: Elaboración Propia).

El botón escanear dispositivos se encuentra asociado al fragmento de código que se apreciar en la ilustración 10. Donde se muestra la implementación de la función search_devices, la

cual ejecuta la instrucción `adb devices` en la terminal mediante la función `subprocess.run()`, instrucción que permite obtener el listado de dispositivos conectados al dispositivo de análisis mediante el puerto USB y regresando los datos obtenidos a través de la sentencia `return a`.

```
def search_devices(self):
    data = subprocess.run(['adb', 'devices'],
                          stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    a = data.stdout.decode().split('\n')
    return a
```

Ilustración 10. Implementación de la Función search_devices() (Fuente: Elaboración Propia).

El botón seleccionar hace el llamado de la función `select_devices()`, implementación que se puede apreciar en la ilustración 11. Función que permite ejecutar instrucciones dentro de la Shell del dispositivo tales como: `ro.build.version.release` ; la cual retorna la versión de Android instalada , `ro.build.version.security_patch` ; retorna la versión de la última actualización de seguridad instalada en el dispositivo. Todos estos datos son almacenados en la lista `result=[]` con la sentencia `result.append(vers)` y retornado con la instrucción `return result.`

```
def selected_device(self, serie):
    result=[]
    data = subprocess.run(['adb', '-s', serie, 'shell', 'getprop',
                           'ro.build.version.release'], stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    vers = data.stdout.decode()
    data = subprocess.run(['adb', '-s', serie, 'shell', 'getprop',
                           'ro.build.version.security_patch'], stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    patch = data.stdout.decode()
    data = subprocess.run(['adb', '-s', serie, 'shell', 'getprop',
                           'ro.product.manufacturer'], stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    manuf = data.stdout.decode()
    data = subprocess.run(['adb', '-s', serie, 'shell', 'getprop',
                           'ro.product.model'], stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    model = data.stdout.decode()
    result.append(serie)
    result.append(manuf)
    result.append(model)
    result.append(vers)
    result.append(patch)
    return result
```

Ilustración 11. Implementación de la Función selected_devices() (Fuente: Elaboración Propia).

Una vez seleccionado el dispositivo, se procede con el registro de este en la base de datos de la aplicación para esto solo basta con hacer clic en el botón agregar.

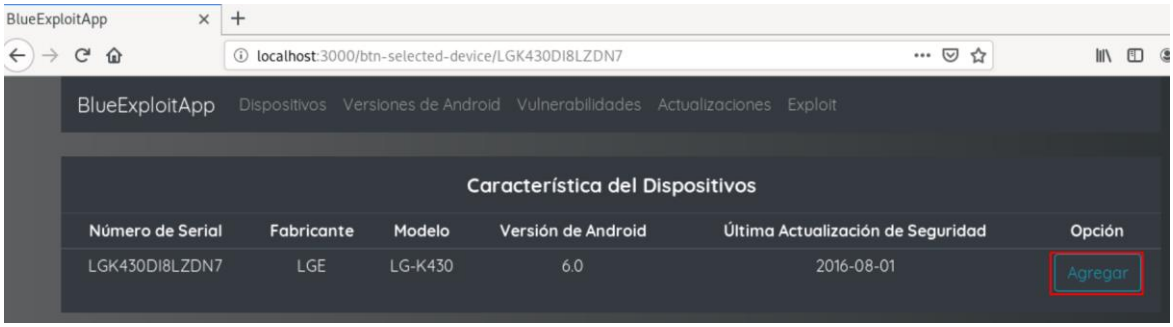

Ilustración 12. Ventana Para Agregar los Dispositivos a la Base de Datos de la Aplicación BlueExploitApp (Fuente: Elaboración Propia).

El botón agregar hace el llamado de la función `btnAddDevices()`, implementación que se puede apreciar en la ilustración 13. Función que recibe los datos del dispositivo analizado mediante el método POST, los cuales son almacenados en la base de datos Devices mediante la siguiente instrucción: `cur.execute('INSERT INTO Devices (idDev, versAndroidDev, versPatchDev, model, fabricante) VALUES (%s,%s,%s,%s,%s)',(d_serie, d_versAndroid, d_versPatch, d_model, d_manufactur,))`.

```
@app.route('/btn-add-device', methods=['POST'])
def btnAddDevices():
    if request.method == 'POST':
        d_serie = request.form['serie'].rstrip()
        d_manufactur = request.form['manufactur'].rstrip()
        d_model = request.form['model'].rstrip()
        d_versAndroid = request.form['versAndroid'].rstrip()
        d_versPatch = request.form['versPatch'].rstrip()
        cur = mysql.connection.cursor()
        cur.execute('SELECT idDev FROM Devices where idDev=(%s)', (d_serie,))
        cur.connection.commit()
        data = cur.fetchall()
        if len(data)is 0:
            cur.execute('INSERT INTO Devices (idDev,versAndroidDev,versPatchDev,model,fabricante)VALUES (%s,%s,%s,%s,%s)',
                        (d_serie, d_versAndroid, d_versPatch, d_model, d_manufactur,))
            cur.connection.commit()
            flash('Dipositivo Añadido Exitosamente')
           # cur.connection.commit()
        else:
            flash('El dispositivo ya se encuentra registrado')
        cur.close()
    data = BlExTools.selected_device(app, d_serie)
    return render_template('analize-device.html', device=data)
```

Ilustración 13. Implementación de la Función btnAddDevices() (Fuente: Elaboración Propia).

Después de agregar el dispositivo se habilitará el botón analizar cómo se observa en la ilustración 14. Al presionar dicho botón se inicia el proceso de identificación de vulnerabilidades.

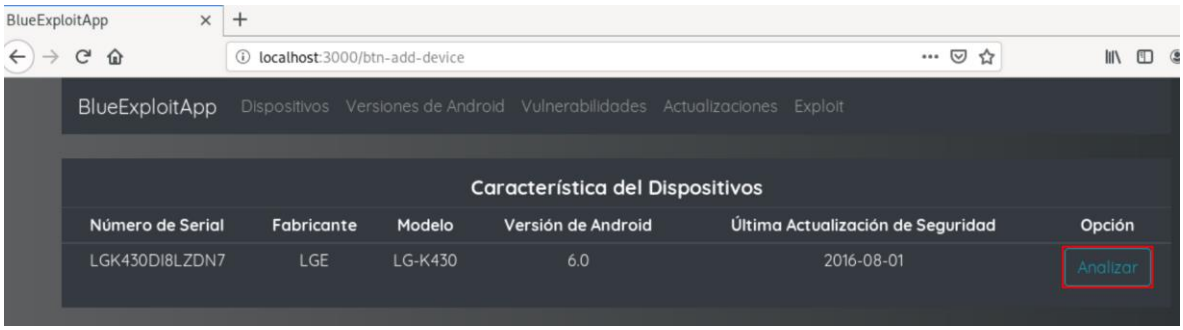

Ilustración 14. Botón analizar habilitado para iniciar la identificación de vulnerabilidades y listar las actualizaciones de seguridad disponibles (Fuente: Elaboración Propia).

El proceso de análisis e identificación de vulnerabilidades es realizado por la función `analizedevices(serie)` implementación que se observa en la ilustración 15. Esta función recibe como parámetro el serial del Dispositivo analizado, parámetro que es pasado a la función consultDevices() implementación que se muestra en la ilustración 16; la cual realiza una consulta en la tabla Devices y retorna los datos obtenidos del dispositivo en cuestión.

Una vez que se obtiene los datos del dispositivo se procede realizar una consulta entre el dispositivo, las vulnerabilidades y actualizaciones de seguridad, obteniendo como resultado las actualizaciones y vulnerabilidades que afectan al dispositivo ver ilustración 17.

```
@app.route('/btn-analize-device/<string:serie>')
def analizeDevices(serie):
    device = consultDevices(serie)
    likeString = "%"+device[1].rstrip()+"%"
    print(likeString)
    cur = mysql.connection.cursor()
    cur.execute('SELECT pat.idPatch, dev.versAndroidDev FROM PatchUpdate pat INNER JOIN Devices dev on(dev.versPatchDev <= pat.idPatch)
                WHERE dev.idDev= % s and pat.versAndroidUpd LIKE % s', (serie, likeString,))
    patch = cur.fetchall()
    cur.close()
    cur = mysql.connection.cursor()
    cur.execute('SELECT vul.cve, exp.typeExploit, exp.scriptRoot  FROM PatchUpdate pat INNER JOIN Devices dev on
                (dev.versPatchDev <= pat.idPatch) INNER JOIN Vulnerabilitys vul ON(pat.idPatch=vul.versUpdate)
                INNER JOIN Exploit exp ON (exp.cveExploit=vul.cve) WHERE dev.idDev= % s and pat.versAndroidUpd LIKE % s', (serie, likeString,))
    vulnerability = cur.fetchall()
    cur.close()
    return render_template('detail-analize-device.html', device=device, patch=patch, vulnerability=vulnerability)
```

Ilustración 15. Implementación de la Función analizeDevices() (Fuente: Elaboración Propia).

Una vez finalizada la identificación de vulnerabilidades se mostrarán dos tablas, una con las actualizaciones de seguridad disponibles para la versión de android que posee el dispositivo y otra tabla con las vulnerabilidades que le pueden afectar al dispositivo al no contar con dichas actualizaciones.

Ilustración 16. Resultado del Análisis de Vulnerabilidades en el Dispositivo Móvil LG-K430 (Fuente: Elaboración Propia).

Este proceso se repite con el dispositivo 2 obteniendo los siguientes resultados.

Ilustración 17. Conexión del Dispositivo 2 (Moto G[5]) y el Dispositivo Portable (Raspberry Pi 4) (Fuente: Elaboración Propia).

Ilustración 18. Resultado del análisis del dispositivo 2 (Moto G[5]) (Fuente: Elaboración Propia).

6.3.3 Instalación de las actualizaciones de seguridad.

Una vez finalizado el proceso de identificación de vulnerabilidades en los dispositivos con la herramienta BlueExploitApp, se obtiene un listado de actualizaciones disponibles teniendo como referencia la versión de android instalada y el fabricante de este dispositivo ya que no todos los modelos alcanzan a ser actualizados.

Inicialmente verificamos que los dispositivos cuenten con conexión a internet a través de la red wifi.

6.3.3.1 Instalación en el Dispositivo 1:

Seguidamente tomamos el dispositivo 1 (LG-K430) y procedemos a instalar las actualizaciones de seguridad para esto se ingresa al menú principal y se escoge la opción ajustes, una vez ingresado en esta se ingresa en la opción General, seguidamente se escoge la opción Acerca del teléfono y dentro de esta la opción centro de actualización donde se escogerá la opción Actualización de software y por último comprobar si hay actualizaciones.

En esta última opción se mostrarán las versiones de android disponibles para el dispositivo, se aceptan los términos, condiciones y se procede con la instalación. Se repite este proceso hasta que se instale la última actualización disponible.

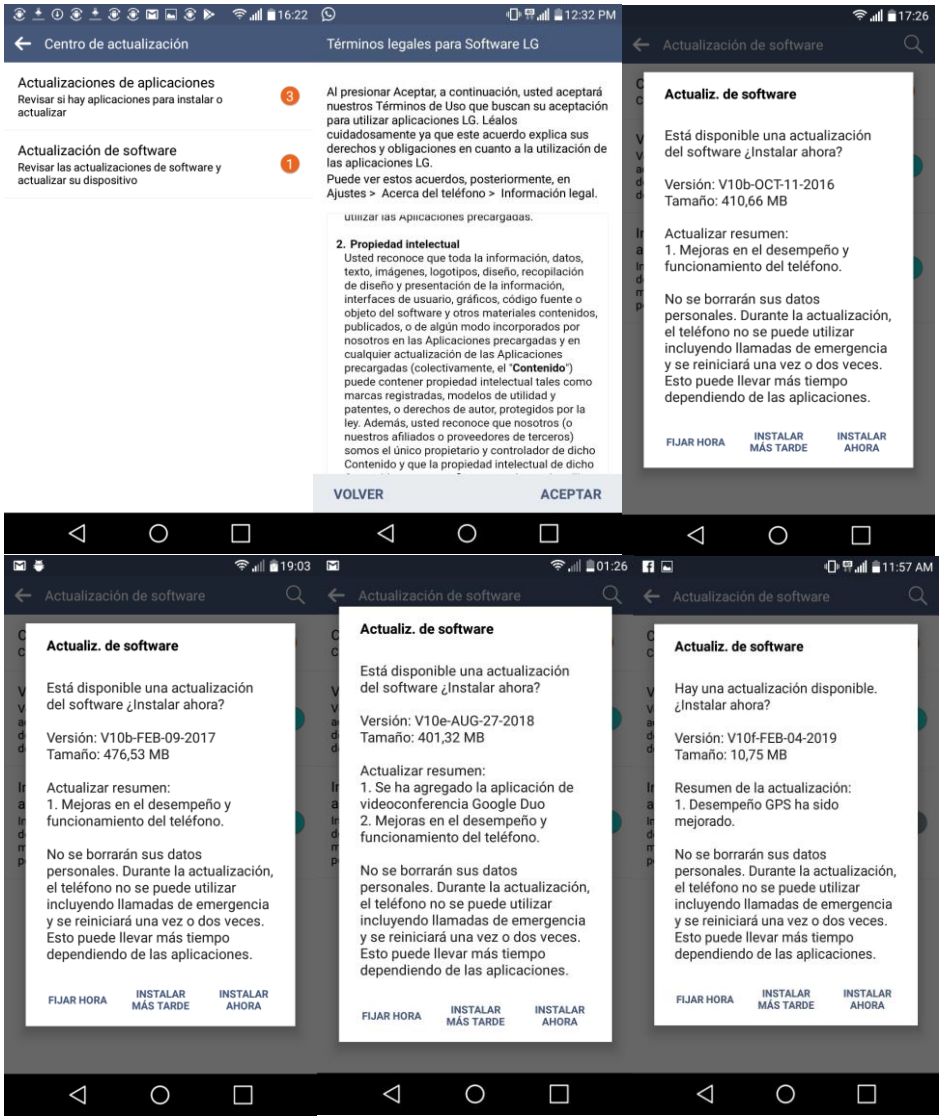

Ilustración 19. Instalación de Actualizaciones de Seguridad en el Dispositivo Móvil LG-K430 (Fuente: Elaboración Propia).

6.3.3.1 Instalación en el Dispositivo 2:

Continuando con el proceso de instalación de los Patch de seguridad en el dispositivo 2 (Moto G[5]), se ingresa al menú principal, luego a la opción configuración, una vez ingresado en esta se escoge la opción Acerca del teléfono y por último Actualización del sistema.

En esta última opción se mostrarán las versiones de android disponibles para el dispositivo, al igual que la última actualización de seguridad.

Se aceptan los términos, condiciones y se procede con la instalación. Se repite este proceso hasta que no sea posible instalar nuevas actualizaciones.

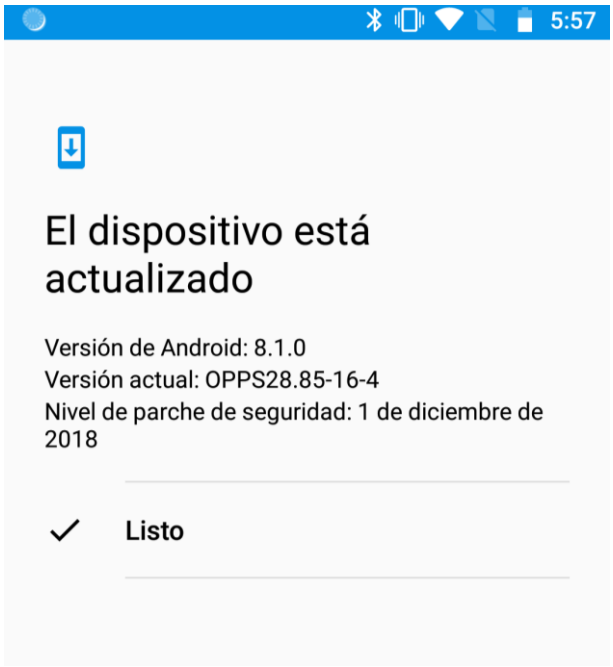

Ilustración 20. Ultima Actualización de Seguridad Disponible para el Dispositivo Móvil Moto G[5] (Fuente: Elaboración Propia).

Cabe recordar que algunos modelos no cuentan con soporte de actualización automático a través de Google Play por lo que la instalación de las actualizaciones debe hacerse de manera manual para esto basta con identificar la actualización de seguridad que corrige la vulnerabilidad que afecta el dispositivo móvil, para esto se tendrá como referencia la información plasmada en la tabla 5.

6.3.4 Validación de las actualizaciones de seguridad.

En el proceso de validación de las actualizaciones de seguridad se dividirá en dos fases con el fin de tener una prueba control de la existencia de dicha vulnerabilidad en el dispositivo analizado: Fase 1. Explotación de la vulnerabilidad en los dispositivos sin actualización de seguridad; Fase 2. Explotación de la vulnerabilidad en los dispositivos con la última actualización de seguridad disponible para el dispositivo.

El proceso de explotación de vulnerabilidad se llevará a cabo aplicando los principios de un pentesting de caja blanca; proceso en el cual se cuenta con la información y permisos necesarios para llevar a cabo el analís en cada uno de los dispositivos, facilitando la identificación y explotación de vulnerabilidades que dicho dispositivo pueda presentar.

6.3.4.1 Explotación de vulnerabilidad sin actualización de seguridad.

Inicialmente buscamos el Exploit dispuesto para comprobar la existencia de la vulnerabilidad a analizar CVE-2017-0785, ingresando a la siguiente dirección web https://www.exploit-db.com/exploits/44555, en esta descargamos el Exploit como se muestra en la ilustración 14.

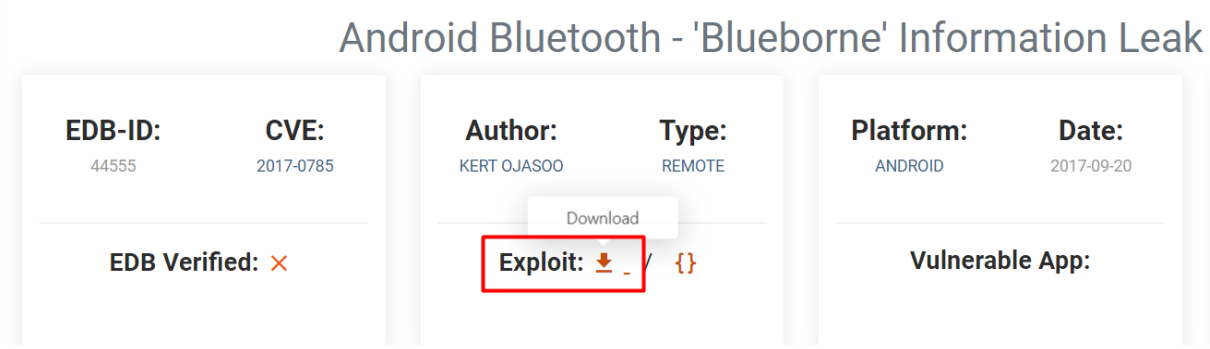

Ilustración 21. Descarga del Exploit que Demuestra la Existencia de la Vulnerabilidad CVE-2017-0785 (Fuente: Elaboración Propia).

Una vez finalizada la descarga, se continua con la preparación del entorno de pruebas para esto es necesario instalar los siguientes repositorios en el dispositivo portable:

```
sudo apt-get install bluetooth libbluetooth-dev
sudo pip install pybluez
sudo pip install pwntools
```

Siguiendo la metodología de caja blanca abrimos una terminal en el dispositivo portable y ejecutamos los siguientes comandos:

`Hciconfig`, este primer comando muestra la configuración bluetooth de la máquina con la cual se realizará todo el proceso de explotación de la vulnerabilidad CVE-2017-0785, en los dispositivos 1 y 2, permitiendo verificar que este se encuentre encendido de lo contrario se utilizará el siguiente comando **`hciconfig hci0 up`**, el cual levanta la interfaz del adaptador bluetooth.

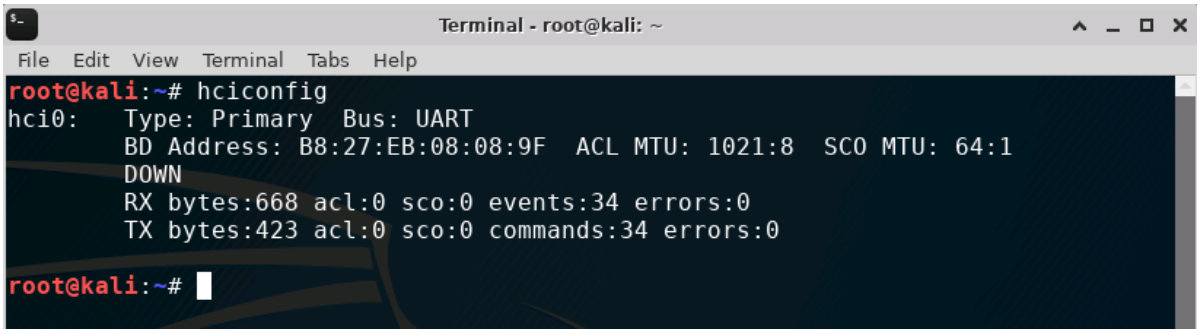

Ilustración 22. Verificación y Encendido del Adaptador Bluetooth en la Máquina Atacante. (Fuente: Elaboración Propia).

Después de validar y encender el dispositivo procedemos a escáner la dirección Mac del dispositivo objetivo, para esto se utiliza el comando **`hcitool scan`**, obteniendo la dirección MAC de los dispositivos que se encuentran disponibles.

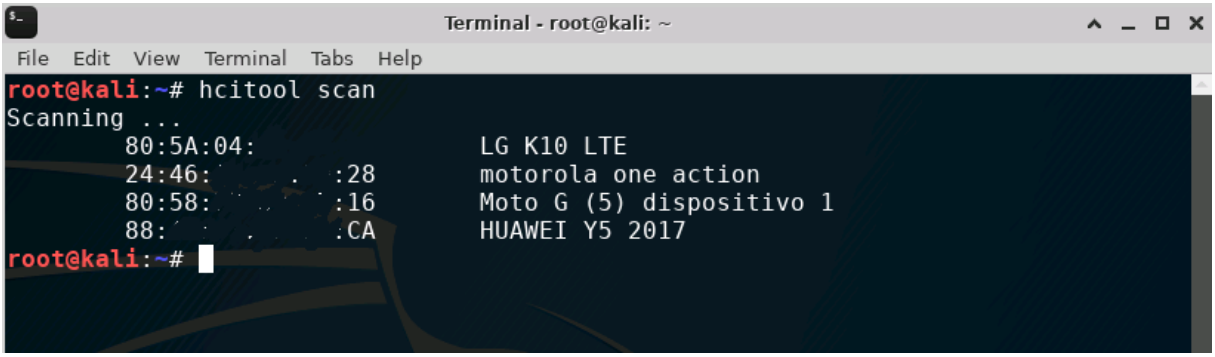

Ilustración 23. Lista de Direcciones MAC de los Dispositivos Móviles con el Adaptador Bluetooth Encendido (Fuente: Elaboración Propia).

Luego abrimos una nueva terminal y accedemos al directorio donde se encuentra el Exploit de la vulnerabilidad CVE-2017-0785. Para llevara a cabo su ejecución se debe asignar como parámetro TARGET la dirección MAC del dispositivo 1 como se muestra en el recuadro de la ilustración 17.

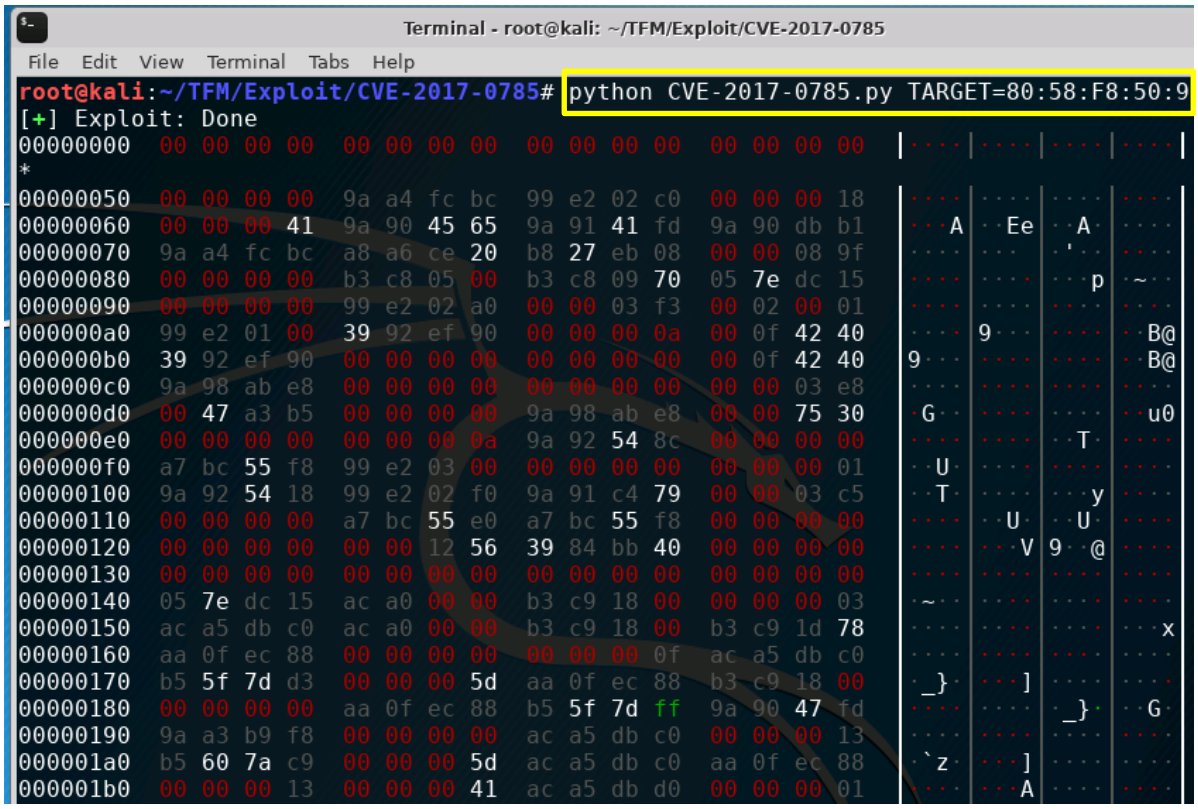

Ilustración 24. Resultado de la Ejecución del Exploit CVE-2017-0785 en el Dispositivo 1 (Fuente: Elaboración Propia).

Al obtener como resultado el conjunto de bits que se observan en la ilustración 17, se logra validar que el dispositivo es afectado por la vulnerabilidad de fuga de información CVE-2017-0785. La cual le permite a un atacante enviar un conjunto de solicitudes diseñadas al servidor SDP, haciendo que este revele bits de memoria como respuesta a esas solicitudes, lo que conlleva a que esta información pueda ser utilizada por el atacante para superar las medidas de seguridad avanzadas y tomar el control del dispositivo (Seri y Vishnepolsky, 2017)

Continuando con la validación se repite el proceso anterior en el dispositivo 2, con la finalidad de verificar si este es afectado por la vulnerabilidad CVE-2017-0785.

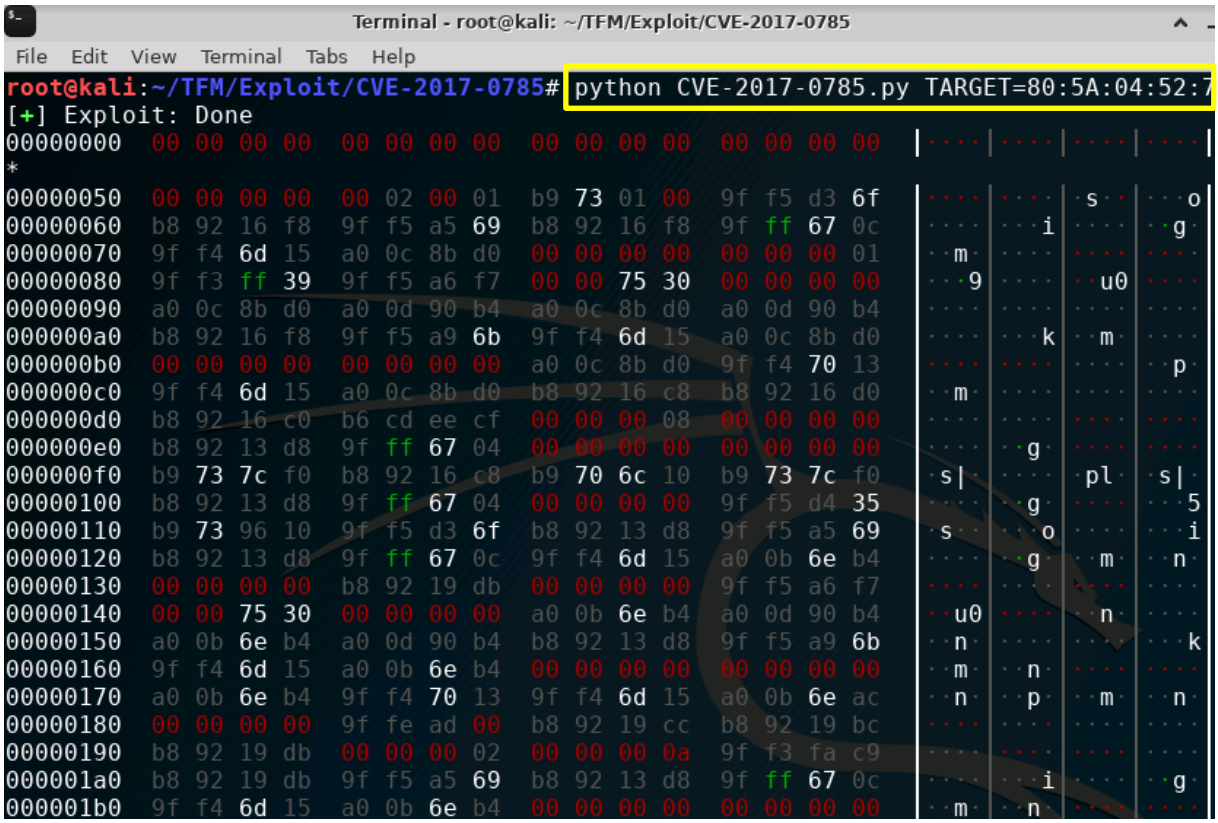

Ilustración 25. Resultado de la Ejecución del Exploit CVE-2017-0785 en el Dispositivo 2 (Fuente: Elaboración Propia).

El resultado que se obtiene después de ejecutar el Exploit ratifica que el dispositivo es afectado por la vulnerabilidad analizada.

6.3.4.2 Explotación de vulnerabilidad después de instalar las actualizaciones de seguridad.

Una vez instaladas la última actualización de seguridad disponible para los dos dispositivos analizados se procede a repetir el proceso de explotación de la vulnerabilidad CVE-2017-0785.

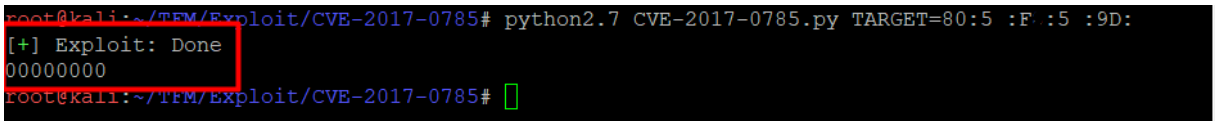

Ilustración 26. Resultado de la Ejecución del Exploit CVE-2017-0785 en el Dispositivo 1 (Fuente: Elaboración Propia).

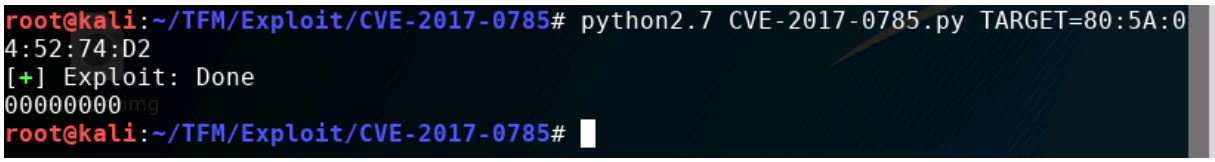

Ilustración 27. Resultado de la Ejecución del Exploit CVE-2017-0785 en el Dispositivo 2 (Fuente: Elaboración Propia).

6.3.5 Informe de los resultados

Durante el desarrollo del piloto se logra evidenciar que la tecnología bluetooth presenta una serie de vulnerabilidades que pueden llegar a afectar gran parte de los usuarios que consumen esta tecnología, en especial los que utilizan dispositivos móviles con sistema operativo android, siendo este un vector de infección o propagación de virus, malware o ejecución de código malicioso. Los cuales pueden ocasionar daños irreparables en un sistema, permitir la extracción de información valiosa para una organización, para obtener una retribución monetaria o satisfacción personal.

Una vez finalizado el análisis, identificación y explotación de una de las vulnerabilidades bluetooth en los dispositivos móviles se procedió a realizar la actualización de los dispositivos con la finalidad de comprobar la efectividad de estas, en este proceso se logra verificar que las actualizaciones corrigen las vulnerabilidades identificadas. Uno de los inconvenientes presentados durante este proceso es la disponibilidad de las actualizaciones de seguridad que brindan los fabricantes para algunos modelos y versiones de android, lo que se convierte en una brecha de seguridad al momento de incluir estos dispositivos como herramientas de trabajo dentro de una organización. A diferencia de los dispositivos que reciben directamente las actualizaciones emitidas por Google ya dichas actualizaciones se pueden instalar de manera oportuna.

También es importante resaltar que la instalación de ROM modificadas puede presentar un problema al momento de recibir actualizaciones de seguridad por parte del servicio de actualizaciones del dispositivo, convirtiéndose en un dispositivo inseguro.

7 Conclusiones y trabajos futuros

7.1 Conclusiones

A continuación, se analizará el cumplimiento de cada uno de los objetivos establecidos en el inciso 3.2:

a. Determinar las vulnerabilidades críticas que presenta la tecnología bluetooth en la plataforma Android.

Se logran identificar una serie de vulnerabilidades bluetooth que afectan a los dispositivos con tecnología bluetooth mediante la consulta de estas en la base de datos de vulnerabilidad nacional NVD, la cuales fueron analizadas y clasificadas según el sistema de puntuación de vulnerabilidades CVSS v3 y CVSS v2 con puntajes alto y crítico, y solo aquellas que afectan a los dispositivos con sistema android, de la cual se obtiene una tabla de vulnerabilidades que fueron pieza clave para el desarrollo de este piloto, se puede evidenciar en la tabla 4 y 5.

b. Explotar las vulnerabilidades críticas que presenta la tecnología bluetooth en la plataforma Android después de las actualizaciones de seguridad.

Este proceso se divide en dos fases, la primera es la explotación de vulnerabilidades sin las actualizaciones de seguridad, las cuales sirven como prueba control sobre el funcionamiento del Exploit a utilizar, tras la explotación se procede a la actualización de los dispositivos con las actualizaciones ofrecidas por sus fabricantes. Se procede a la explotación de la vulnerabilidad CVE-2017-0785 proceso del cual no se obtiene resultados satisfactorios ya que la actualización corrigió de manera efectiva dicha vulnerabilidad todo este preso se puede evidenciar en el inciso 4.3.4.

c. Correlacionar las vulnerabilidades que presenta la tecnología bluetooth en la plataforma Android antes y después de las actualizaciones.

Después de realizar estos dos procesos se puede concluir que las actualizaciones de seguridad cumplen un papel importante en el mantenimiento de la seguridad de los dispositivos ya que logran corregir las vulnerabilidades repostadas en NVD, en este proceso se logra identificar una problemática por parte de los fabricantes ya que algunos no logran lanzar dichas actualizaciones de manera oportuna para algunas versiones y modelos de celulares en especial las versiones más antiguas.

d. Construir una Whitelist de dispositivos seguros que puedan ser utilizados en la modalidad de trabajo (Bring Your Own Devices).

Como resultado del piloto se crea una Whitelist de dispositivo seguros, que pueden ser utilizados en la modalidad de trabajo Bring Your Own Devices, ya que la instalación de los Patch de seguridad solventa las vulnerabilidades presentes en dicha versión de android, es muy importante resaltar que solo se realizaron pruebas de vulnerabilidades bluetooth en varios dispositivos, comprobando la efectividad de las actualizaciones de seguridad.

7.2 Trabajos futuros

Analizando los procesos realizados y la metodología planteada para realizar el análisis de cada uno de los dispositivos se llega a la conclusión de que es una labor un poco tediosa ya que es necesario realizar una serie de procedimientos monótonos los cuales se pueden automatizar de tal manera que se facilite esta labor, a raíz de estos se plantea el desarrollo de una herramienta tecnológica que pueda ser tenida en cuenta por el CIO de la organización al momento de desarrollar e implantar políticas BYOD.

La implantación de esta política no es una tarea fácil para el CISO ya que debe encargarse de la seguridad de dispositivos que son de usos personal, por lo que se recomienda la implantación de plataformas o herramientas tecnológicas que permitan la gestión de estos de tal manera que se pueda llevar un control de los dispositivos seguros e inseguros y que a su vez permita gestionar la conexión de los dispositivos seguros con los diferentes servicios o plataformas con la que cuenta la empresa facilitando la labor del CISO.

Partiendo de la problemática expresada en el párrafo anterior se tiene como iniciativa el diseño de una herramienta llamada BlueExploitApp que permita el análisis, gestión y administración de los dispositivos empleados en BYOD, actualmente es una versión de prueba que permite identificar las vulnerabilidades que se trataron en el desarrollo de este piloto pero con la capacidad de poder identificar cualquiera de las vulnerabilidades reportadas por el NIST, además permite almacenar en su base de datos los dispositivos analizados de tal manera que se le pueda hacer seguimiento a cada dispositivo, inicialmente está orientada a dispositivos android pero puede llegar a soportar dispositivos con sistema iOs. También permite identificar las vulnerabilidades que pueden afectar al dispositivo y las actualizaciones que se deberían instalar para solucionar dichas vulnerabilidades.

Se pretende continuar con el desarrollo de esta herramienta, de tal manera que permita instalar directamente en el dispositivo las actualizaciones que requiera y adicionalmente esta permita realizar la ejecución de Exploit que ratifiquen la efectividad del parche de seguridad instalado, además de gestionar los servicios con los que cuenta la organización y poder dar acceso solo a los dispositivos seguros.

8 Referentes bibliográficos

Afreen, R. (2014). Bring Your Own Device (BYOD) in Higher Education: Opportunities and Challenges. *International Journal of Emerging Trends & Technology in Computer Science*, *3*(1), 5.

Akhayad, Y. (2016). *Bluetooth 4.0 Low Energy: Análisis de las Prestaciones y Aplicaciones Para la Automoción* [Universitat Politècnica de Catalunya]. https://upcommons.upc.edu/handle/2117/82702

Albahar, M. A. (2017). *Bluetooth Pairing Security Threats and Countermeasures*. https://www.researchgate.net/publication/323956636

Android Partner Docs. (2019). *About the Android Open Source Project*. Android Open Source Project. https://source.android.com/?hl=es

Avila, L., & Reyes, C. (2017). Revisión estado del Arte de la tecnología Bluetooth. *Revista Investigación y Desarrollo en TIC*, *3*(2), 7.

Bisdikian, C. (2001). An overview of the Bluetooth wireless technology. *IEEE Communications Magazine*, *39*(12), 86-94. https://doi.org/10.1109/35.968817

Castellano, A. R. (2012). Bluetooth. Introducción a su Funcionamiento. *Universidad Pontificia Comillas, Madrid*, *1*(1), 1-16.

Darroudi, S. M., & Gomez, C. (2017). Bluetooth Low Energy Mesh Networks: A Survey. *Sensors*, *17*(7), 1467. https://doi.org/10.3390/s17071467

Duque, F. B., & Salazar, L. A. P. (2019). *Desarrollo de una Aplicación que Permita Disminuir las Vulnerabilidades en Conexiones Bluetooth en Sistemas Operativos Android* [Universidad Distrital Francisco José de Caldas]. http://repository.udistrital.edu.co/handle/11349/22419

g0tmi1k. (2019). *Documentación de Kali Linux*. https://www.kali.org/docs/introduction/what-is-kali-linux/

INCIBE-CERT. (2015, julio 21). *Métricas de evaluación de vulnerabilidades: CVSS 3.0*. https://www.incibe-cert.es/blog/cvss3-0

Morales Pedro, R. (2011). *Bluetooth v4.0: La futura solució inalámbrica de baix consum* [Bachelor's thesis]. Universitat Politècnica de Catalunya.

Moreno, F., & Ramírez, E. (2017). *Algoritmos de Visión por Computador para un SBC*. 13.

Muller, N. J. (2002). *Tecnología Bluetooth: Nathan J. Muller; Traducción y revisión técnica VUELAPLUMA, SL*. McGraw-Hill.

Nateq Be-Nazir Ibn Mina, Tarique, M., & Tarique, M. (2012). Bluetooth Security Threats and Solutions: A Survey. *International Journal of Distributed and Parallel Systems*, *3*. https://doi.org/10.5121/ijdps.2012.3110

Outeiriño, F. J. B., Fernández, J., Roldán, M., & Peyrona, J. (2004). Comunicación Inalámbrica con Bluetooth. *Técnica Industrial*, *1*, 18-23.

Raspberry Pi Foundation. (s. f.). *Raspberry Pi Foundation-about*. https://www.raspberrypi.org/about/

Rocha, É. S., & Bruno, G. G. E. (2006). *Estudo e Análise de Vulnerabilidades de Segurança na Tecnologia Bluetooth*. 6.

Romero Castro, M. I., Figueroa Morán, G. L., Vera Navarrete, D. S., Álava Cruzatty, J. E., Parrales Anzúles, G. R., Álava Mero, C. J., Murillo Quimiz, Á. L., & Castillo Merino, M. A. (2018). *Introducción a la Seguridad Informática y el Análisis de Vulnerabilidades* (1.ª ed.). Editorial Científica 3Ciencias. https://doi.org/10.17993/IngyTec.2018.46

Seri, B., & Vishnepolsky, G. (2017). *BlueBorne Technical White Pape*. https://go.armis.com/hubfs/BlueBorne%20Technical%20White%20Paper-1.pdf

Sparacino, G. L. (2003). Tecnología Inalámbrica Bluetooth Sobre los Servicios de Comunicaciones en los Ámbitos Social y Empresarial. *Telématique*, *2*(2), 36-49.

Vergara, S. (2008). *Tecnología Bluetooth*. Instituto Politécnico Nacional.

Yiwen Wu. (2019, octubre 30). *Strategy Analytics: Global Smartphone Shipments Return to 2 Percent Growth in Q3 2019 | Strategy Analytics Online Newsroom*. https://news.strategyanalytics.com/press-release/devices/strategy-analytics-global-smartphone-shipments-return-2-percent-growth-q3-2019

Zafra, G., & Luis, J. (2017). *Introducción al Pentesting*. http://diposit.ub.edu/dspace/handle/2445/124085

9 Anexos

9.1 Anexo 1: Instalación de Kali Linux en Raspberry Pi 4 modelo B

Para iniciar el proceso de instalación nos dirigimos al siguiente enlace https://www.offensive-security.com/kali-linux-arm-images/ donde se podrá descargar la distribución de Kali Linux compatible con el modelo de Raspberry a utilizar en el piloto.

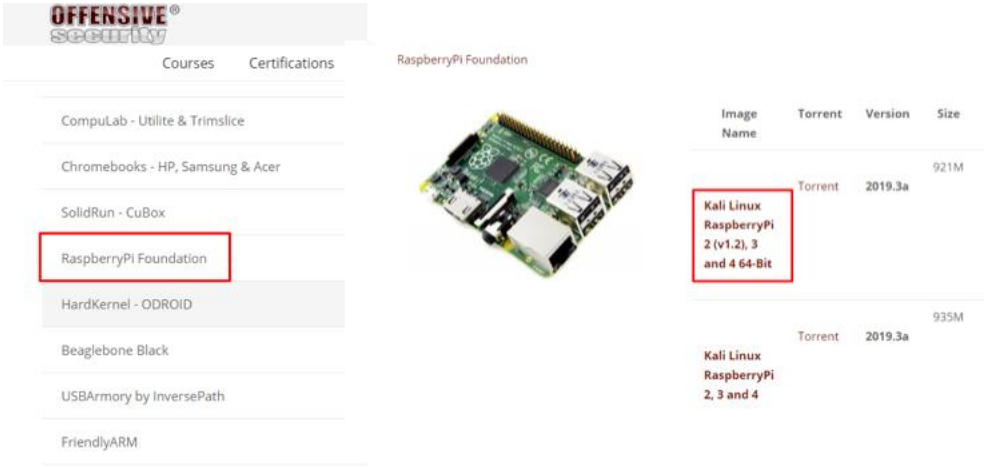

Ilustración 28. Pasos Para Descargar la Imagen de Kali Linux (Fuente: Elaboración Propia).

Luego de descargar la imagen de Kali Linux se procede con el copiado de está en la memoria SD, para eso se utilizara el programa Win32diskimager que podemos descargar desde el siguiente enlace https://sourceforge.net/projects/win32diskimager/files/latest/download. Programa que se instalará en el equipo 1.

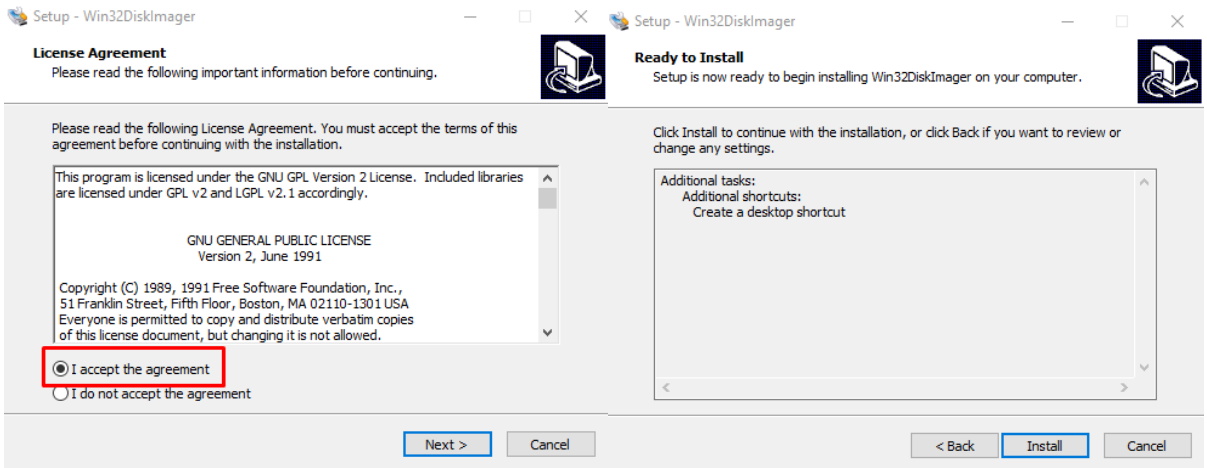

Ilustración 29. Instalación del Programa Win32diskimager en Windows 10 (Fuente: Elaboración Propia).

Seguidamente, se procede con la preparación del dispositivo SSD en el cual se realizar el copiado de la imagen de Kali Linux utilizando los siguientes pasos:

1. Seleccionar la ruta donde se encuentra almacenada la imagen de Kali Linux.
2. Seleccionar la unidad donde se copiará la imagen de Kali Linux.
3. Hacer clic en la opción write, para iniciar el copiado.

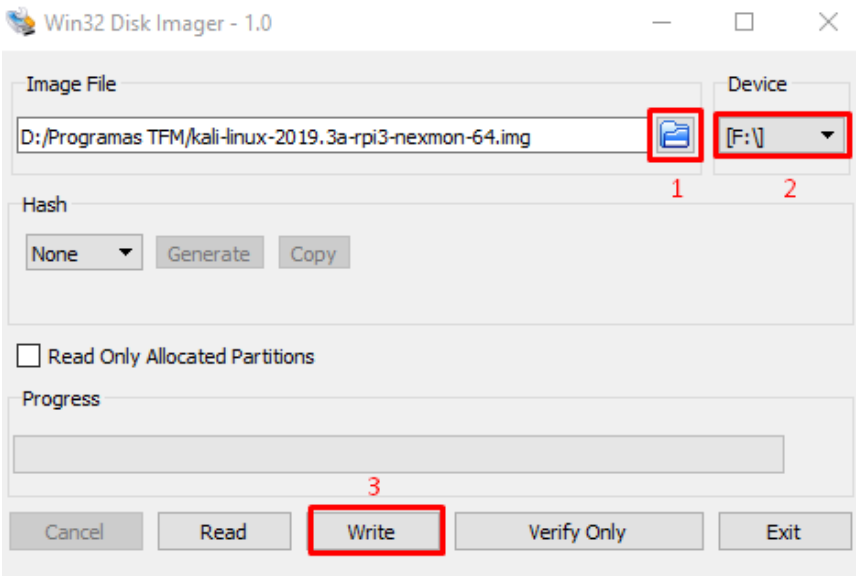

Ilustración 30. Pasos Para Copiar la Imagen y Crear el Arranque de Kali Linux en la Memoria SSD (Fuente: Elaboración Propia).

Finalizado el proceso de copiado se inserta la tarjeta SSD en la Raspberry Pi, además se deben conectar los periféricos de entrada/salida necesarios para la configuración inicial (Ratón, teclado y monitor), por último, se conecta el cable de alimentación eléctrica para que se cargue el sistema operativo.

Una vez cargado el sistema operativo se procede con el inicio de sesión. Para esto se deben ingresar las credenciales de usuario: ***root*** y contraseña: ***toor***.

Para finalizar este proceso se actualizan los repositorios y el sistema operativo Kali Linux utilizando el siguiente comando ***apt update && apt -y full-upgrade -y***

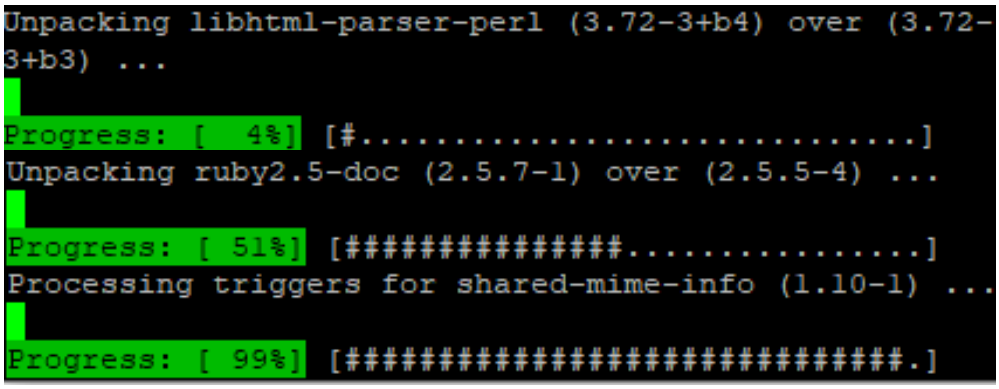

Ilustración 31. Actualización del Sistema Operativo Kali Linux (Fuente: Elaboración Propia).

Para la configuración de la pantalla se abre la terminal y digitamos el siguiente comando ***kalipi-tfl-config,*** en el menú de opciones que se despliega se debe escoger la opción 1, seguidamente aparecerán un listado de referencia, se debe escoger la adecuada y así evitar inconvenientes con los controladores que se instalan.

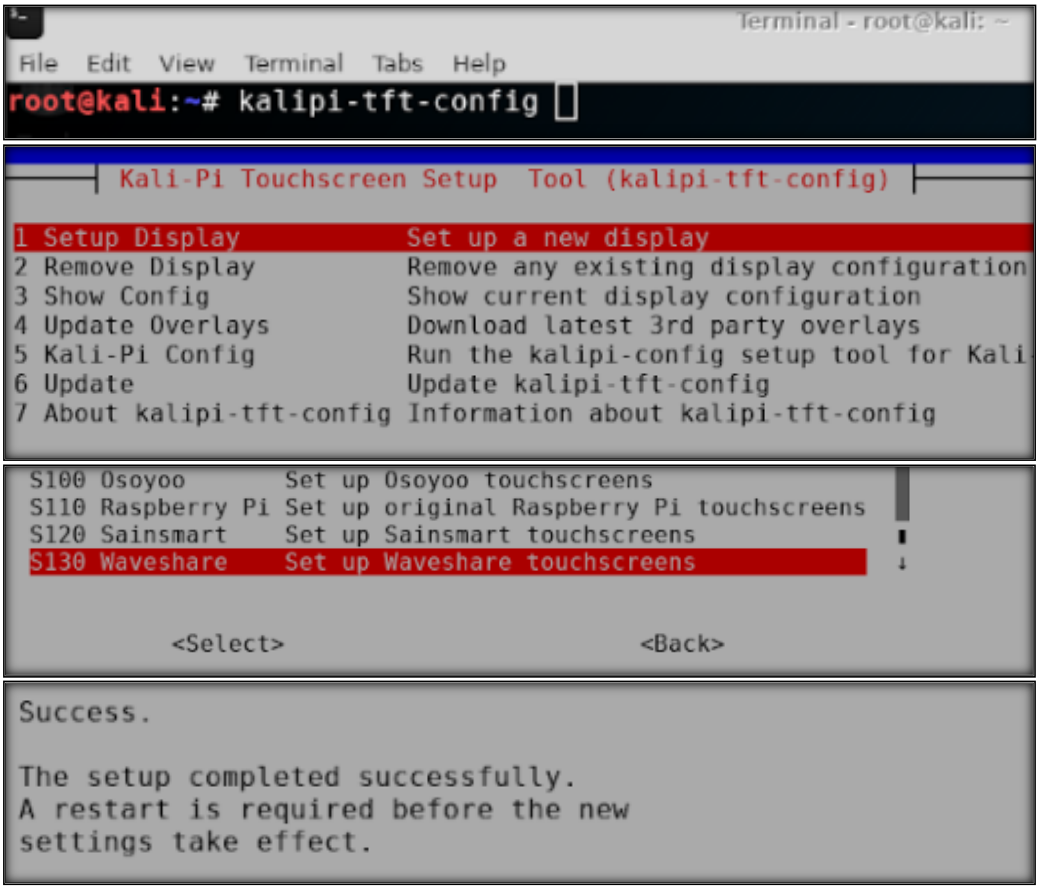

Ilustración 32. Secuencia de Pasos Para la Instalación de los Controladores de un Display Touch Screen de 3.2 Pulgadas en una Raspberry pi 4 Con Sistema Operativo Kali Linux (Fuente: Elaboración Propia)

9.2 Anexo2: Instalación y manual de uso de la herramienta BlueExploitApp.

La herramienta BlueExploitApp actualmente solo está disponible para la distribución Kali Linux.

El proceso de instalación inicia con la clonación del repositorio GitHub https://github.com/pater8715/TFM.git, el cual se puede hacer desde la terminal o directamente de la página web.

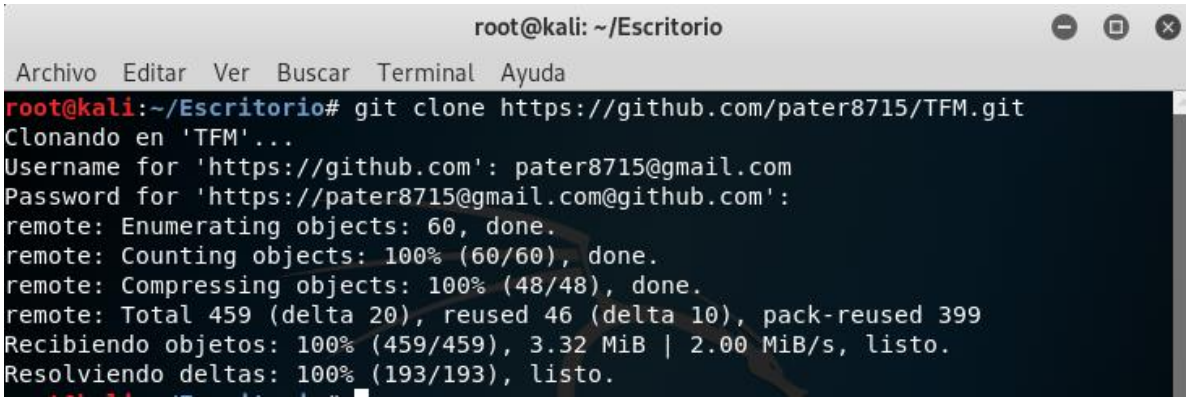

Ilustración 33. Clonación del Repositorio de la Herramienta BlueExploitApp utilizando la Terminal (Fuente: Elaboración Propia)

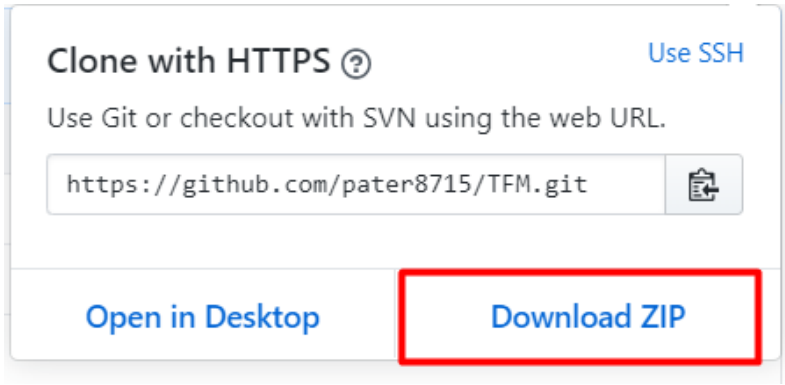

Ilustración 34. Descarga del Repositorio de la Herramienta BlueExploitApp (Fuente: Elaboración Propia)

Una vez clonado el repositorio procedemos a ingresar en el directorio FTM/Servidor/, en este directorio encontraremos dos scripts de Python que permite la instalación de las librerías necesarias para la ejecución de la herramienta BlueExploitApp, si se desea instalar en un entorno virtualizado se debe utilizar el siguiente comando `python3 instalarservidor.py` o si se desea instalar en una distribución ARM se debe ejecutar el siguiente comando `python3 installServerArm.py`.

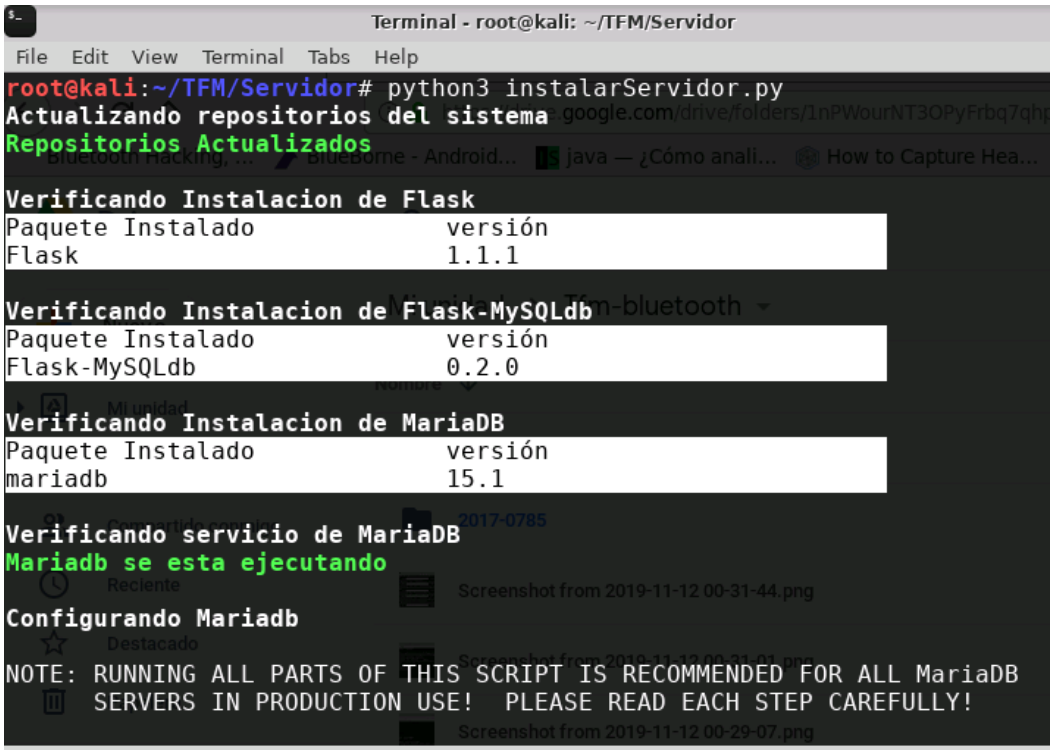

Ilustración 35. Instalación de las Librerías Necesarias para Ejecutar la Herramienta BlueExploitApp y la Configuración del Servidor de Base de Datos (Fuente: Elaboración Propia)

Luego se debe ingresar en siguiente ruta para instalar la base de datos que utilizara la herramienta /FTM/Instalador, dentro del directorio se ejecuta el script de instalación con el siguiente comando `python3 database_install.py`

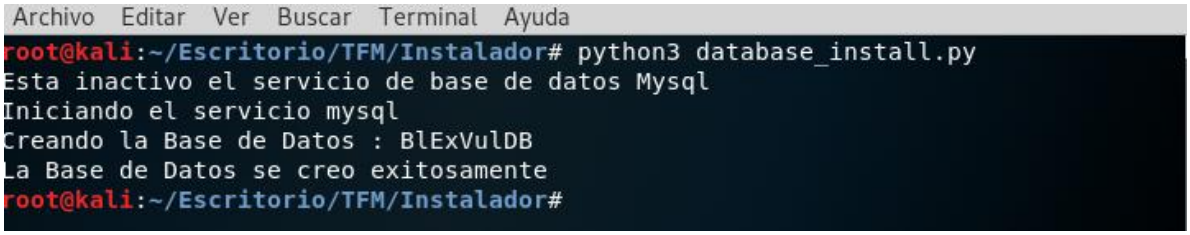

Ilustración 36. Instalación de la Base de Datos de la Herramienta BlueExploitApp (Fuente: Elaboración Propia)

Después de realizar la instalación se debe ingresar al directorio /TFM/BlueExploitApp/ y ejecutar el siguiente comando `python3 App.py` para iniciar la ejecución de la herramienta.

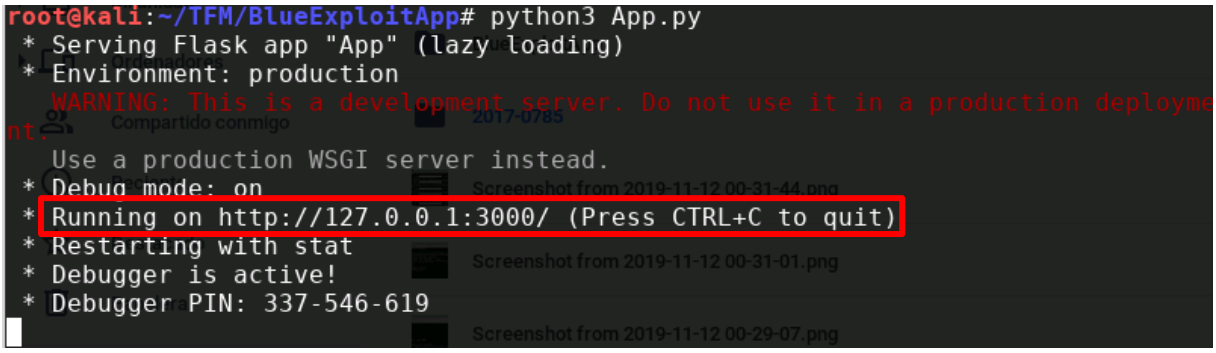

Ilustración 37. Puesta en Marcha de la Herramienta BlueExploitApp y Ruta de Acceso (Fuente: Elaboración Propia)

Una vez ejecutada la herramienta, se abre el navegador web y se escribe la dirección que se resalta en la ilustración 33.

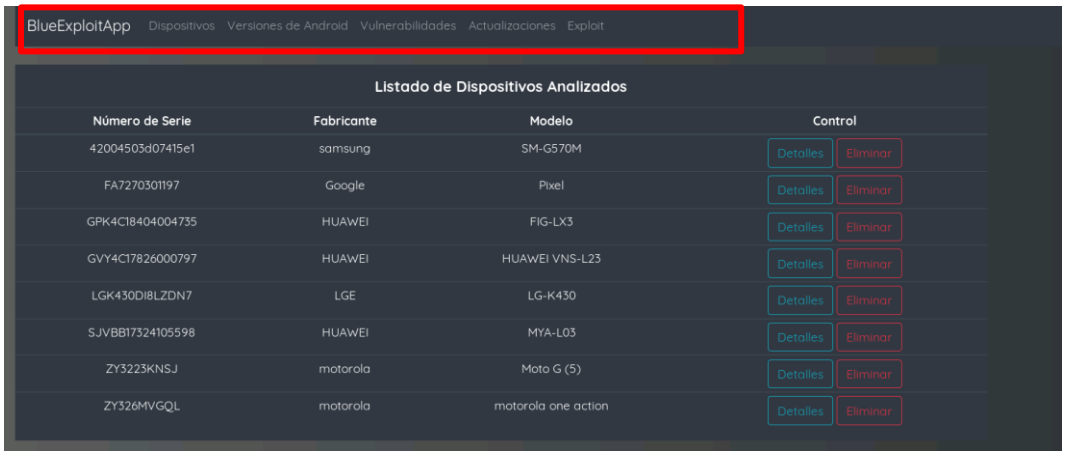

Ilustración 38. Página Principal de la Herramienta BlueExploitApp (Fuente: Elaboración Propia)

En esta página se muestra un listado de los dispositivos analizados, además cuenta con un menú de opciones que permite al usuario realizar diferentes procesos tales como:

- Dispositivo: en esta opción el usuario puede escanear los dispositivos conectados al dispositivo portable mediante cable USB y realizar el respectivo análisis de vulnerabilidades.

Ilustración 39. Lista de dispositivos disponibles para ser analizados (Fuente: Elaboración Propia)

- Versiones de android: en esta ventana se pueden ir actualizando la base de datos de las versiones de android, además se visualizan las versiones almacenadas en la base de datos habilitando la opción de eliminar.

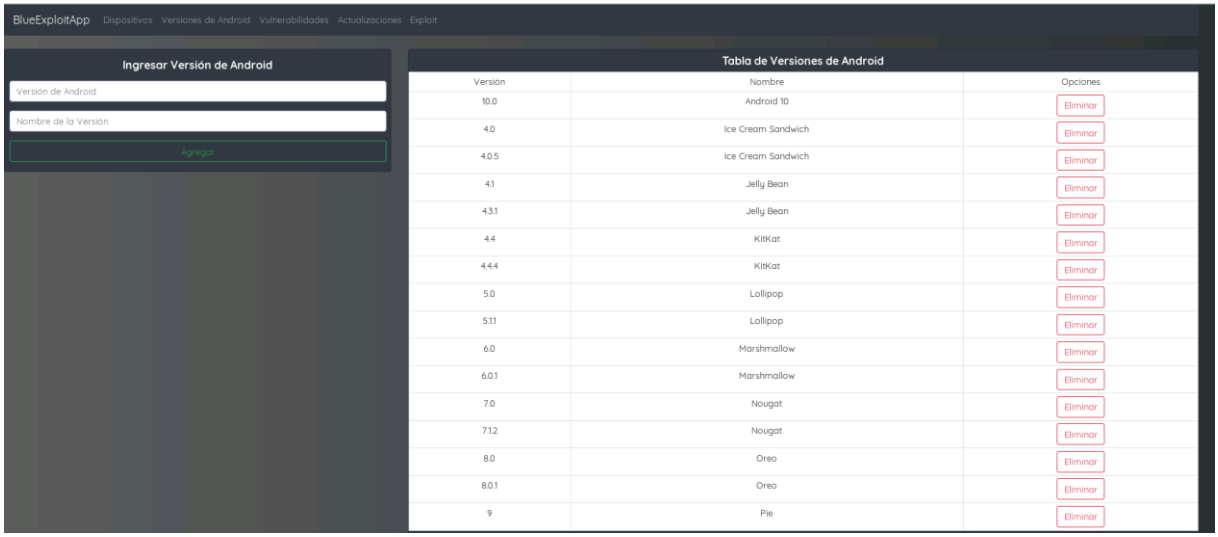

Ilustración 40. Ventana Versiones de Android y Lista de versiones de Android (Fuente: Elaboración Propia)

- Vulnerabilidades: en esta ventana se pueden ir actualizando la base de datos de las vulnerabilidades que presente la tecnología bluetooth u otro tipo de vulnerabilidades, además se visualizan las vulnerabilidades almacenadas en la base de datos habilitando la opción de eliminar.

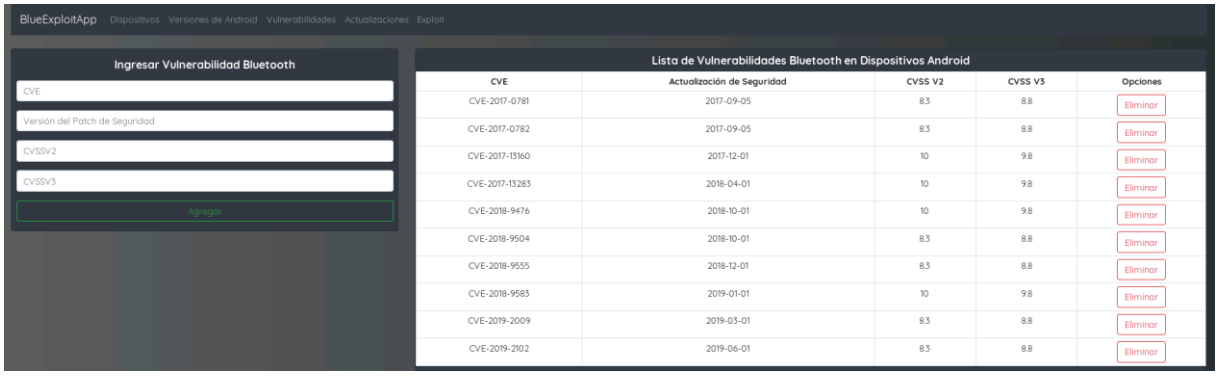

Ilustración 41. Ventana Vulnerabilidades y Lista de Vulnerabilidades Bluetooth que Afectan a los Dispositivos Android (Fuente: Elaboración Propia)

- Actualizaciones: en esta ventana se pueden ir actualizando la base de datos con las actualizaciones disponibles para cada versión de android, además se visualizan las Actualizaciones almacenadas en la base de datos habilitando la opción de eliminar.

Ilustración 42. Ventana Actualizaciones y Lista de Actualizaciones Disponibles por Versión de Android y Fabricante (Fuente: Elaboración Propia)

- Exploit: en esta ventana se pueden ir actualizando la base de datos con las Exploit que puedan evidenciar la presencia de una vulnerabilidad, además se visualizan los Exploit almacenados en la base de datos habilitando la opción de eliminar.

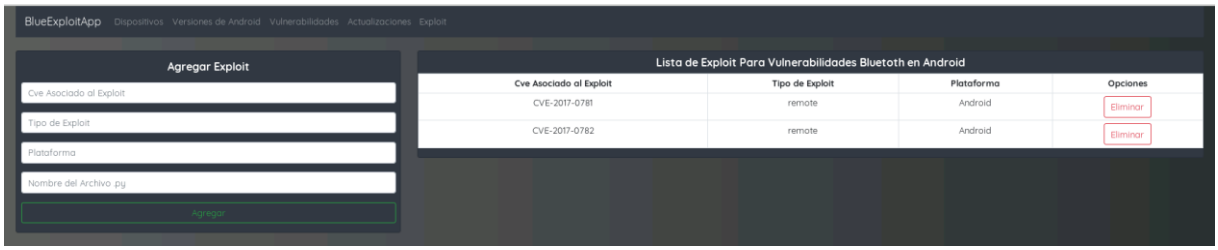

Ilustración 43. Ventana Exploit y Lista Exploit que Evidencia la Vulnerabilidad Detectada. (Fuente: Elaboración Propia)

9.3 Anexo3: Habilitar la opción de depuración USB.

Para habilitar el modo depuración inicialmente se debe habilitar la opción de programador para esto es necesario ir a la opción configuración -> acerca del dispositivo -> número de compilación, presionamos esta última opción 5 veces hasta que aparezca el mensaje ya eres programador.

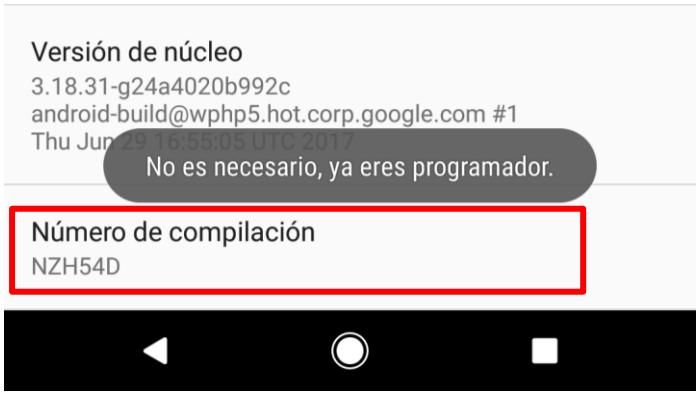

Ilustración 44. Habilitando la Opción de Programador en Android (Fuente: Elaboración propia)

Luego nos dirigimos a la opción configuración -> opciones de programador y habilitamos la opción depuración por USB.

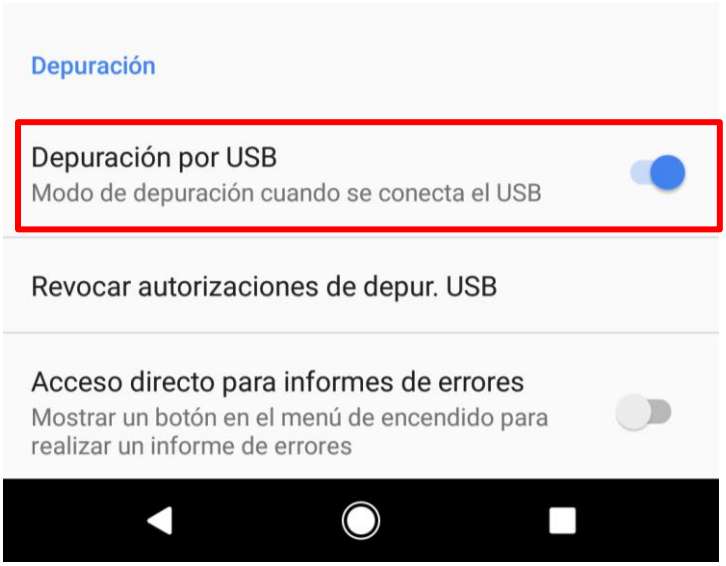

Ilustración 45. Habilitando la Opción de Depuración USB en Android (Fuente: Elaboración Propia)