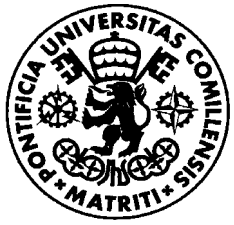


UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO EN INFORMÁTICA

PROYECTO FIN DE CARRERA

SEGURIDAD EN BLUETOOTH

AUTOR: Alberto Moreno Tablado

MADRID, Junio 2006

SEGURIDAD EN BLUETOOTH

Autor: Moreno Tablado, Alberto

Directora: Martínez de Albornoz Torrente, Rosario

Entidad Colaboradora: ICAI – Universidad Pontificia de Comillas

RESUMEN DEL PROYECTO

Bluetooth es la especificación que define un estándar global de comunicaciones inalámbricas para redes de área personal que permite la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia en entornos de comunicaciones móviles y estáticos.

En un entorno que cada día exige una mayor interoperabilidad entre los diferentes equipos existentes en el mercado, la popularidad de Bluetooth se ha visto fuertemente impulsada por su integración en dispositivos de la vida cotidiana como teléfonos móviles y por facilitar la interconexión inalámbrica de periféricos a un ordenador. Así mismo, han surgido nuevos modelos de uso de la tecnología Bluetooth que hacen habitual el empleo de dispositivos de última generación, como los equipos Manos Libres de automóvil, auriculares, módulos GPS, etc.

Desde su lanzamiento, Bluetooth ha sido objeto de estudio por parte de grupos dedicados a la seguridad digital y no han tardado en aparecer las primeras vulnerabilidades. Los primeros ataques se desarrollaron contra los dispositivos Bluetooth más extendidos: los teléfonos móviles. Posteriormente, el estudio se ha ido ampliando a otros tipos de dispositivos, como los equipos Manos Libres.

Los mayores perjudicados con el descubrimiento y publicación de vulnerabilidades en dispositivos Bluetooth son los usuarios propietarios de los mismos, ya que las técnicas de ataque desarrolladas con el fin de explotar estos agujeros de seguridad, pueden afectar a su intimidad, a la confidencialidad de sus conversaciones y a la integridad de sus dispositivos.

El objetivo de este proyecto de fin de carrera ha sido realizar un estudio general de la seguridad en Bluetooth.

En primer lugar, ha sido necesario profundizar en la especificación del estándar de comunicaciones y describir la arquitectura de protocolos, los modelos de uso, los servicios, los perfiles y, por último, los mecanismos de seguridad que ofrece Bluetooth.
A continuación, se han enumerado los distintos dispositivos que incluyen la tecnología Bluetooth disponibles en el mercado y se ha descrito el proceso de interconexión entre ellos a través del protocolo de comunicaciones.

Posteriormente, se ha llevado a cabo una investigación exhaustiva de las vulnerabilidades existentes en dispositivos Bluetooth y se han descrito con detalle las técnicas de ataques desarrolladas para la explotación de estas fallas de seguridad.

Este proyecto se plantea como un estudio práctico sobre una base teórica de la seguridad en Bluetooth. Se han documentado casos prácticos sobre dispositivos Bluetooth reales que han permitido demostrar de forma visible cómo es posible explotar vulnerabilidades de forma simple con ayuda de herramientas, algunas de las cuales se han desarrollado durante este proyecto.

Con el fin de evitar ser víctima de un ataque por medio de dispositivos Bluetooth, se han dictado unas recomendaciones para el buen uso de la tecnología Bluetooth en aquellos equipos que la incorporan. Se trata de normas simples y de aplicación inmediata que deberían formar parte de la conducta habitual de un usuario de dispositivos Bluetooth.

Por último, se ha hecho especial hincapié en el uso de Linux como plataforma avanzada para el establecimiento de comunicaciones Bluetooth, ya que ofrece una pila de protocolos, BlueZ, sencilla de manejar, con multitud de herramientas a disposición del usuario y con un entorno de desarrollo que simplifica la programación de potentes aplicaciones.

BLUETOOTH SECURITY

Author: Moreno Tablado, Alberto

Director: Martínez de Albornoz Torrente, Rosario

Collaborating Organization: ICAI – Universidad Pontificia de Comillas

ABSTRACT

Bluetooth is the specification that defines a global standard for wireless communications in personal area networks. It allows the transmission of voice and data among different equipment through a radio frequency link in mobile and stationary communication environments.

In a world that demands increasing levels of interoperability among the different available pieces of equipment, the popularity of Bluetooth has been strongly driven by its integration in daily use devices, like cell phones and the way it simplifies the wireless interconnection of computer peripheral devices. Last generation devices enabled with Bluetooth technology have been released lately providing end users with hands free equipment for automobile, headsets, GPS modules, etc. Since its inception, Bluetooth has been subject of study by groups dedicated to the investigation of digital security and first vulnerabilities came up very soon. The first attacks were developed against the most extended Bluetooth devices: cell phones. Later, the study has been extended to other kinds of devices, like hands free equipment.

Bluetooth devices users are the real victims in the discovery and publication of vulnerabilities in Bluetooth devices because attacks have been developed in order to exploit these security holes and may affect their privacy, the confidentiality of their conversations and the integrity of their devices.

The objective of this project is to develop a general study of the security issues in Bluetooth.

In the first place, it was convenient to deepen in the specification of the standard of communications and describe the architecture of protocols, the user models, the services, the profiles and, finally, the mechanisms of security implemented in Bluetooth.

Secondly, different Bluetooth-enabled devices have been enumerated and the interconnection process among them through the protocol of communications described.

Later, an exhaustive investigation of the existing vulnerabilities in Bluetooth devices has been carried out and different techniques of attacks have been described with full detail.

This project can be considered as a practical study based on theoretical research of security issues in Bluetooth. Practical experiments on real Bluetooth devices have been documented and they have allowed to demonstrate how easy it is possible to exploit vulnerabilities using tools, some of which have been developed along this project.

Recommendations for the good use of the Bluetooth technology enabled equipment have been dictated with the purpose of avoiding to suffer potential attacks. These are simple norms that would have to be a part of the usual behavior of a Bluetooth devices user.

Finally, this project makes special emphasis in the use of Linux as an advanced platform for Bluetooth communications, since it is based in a protocol stack, BlueZ, easy to handle, with a wide variety of tools to disposition of the user and with development environments that simplifies the programming of powerful applications.

Quería agradecer la dirección de Rosario Martínez, sin si ayuda este proyecto no habría salido adelante. Dedico este proyecto a mi familia y amigos; sus animos y su comprensión han sido de gran apoyo para finalizar mis estudios.
Recuerdos al Bluehack Team, por iniciarme en el estudio de la seguridad en Bluetooth.

Alberto Moreno, 2006

Índice

Capítulo

1

INTRODUCCIÓN

Bluetooth se ha convertido en el estándar de referencia de comunicaciones inalámbricas para redes de área personal que permite la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia.

En un entorno que cada día exige una mayor interoperabilidad entre los diferentes equipos que existen en el mercado, la popularidad de Bluetooth se ha visto fuertemente impulsada por su integración en dispositivos de la vida cotidiana como teléfonos móviles y por facilitar la interconexión inalámbrica de periféricos a un ordenador. Así mismo, han surgido modelos de uso de la tecnología Bluetooth que hacen habitual el empleo de dispositivos de última generación, como los equipos Manos Libres de automóvil y Auriculares, que permiten mantener conversaciones telefónicas con absoluta libertad mientras se conduce un vehículo o se trabaja en la oficina.

Desde su lanzamiento y como cualquier protocolo de comunicaciones, Bluetooth ha sido objeto de estudio por parte de grupos dedicados a la seguridad digital y no han tardado en aparecer las primeras vulnerabilidades. Los primeros ataques se desarrollaron contra los dispositivos Bluetooth más extendidos: los teléfonos móviles. Posteriormente, el estudio se ha ido ampliando a otros tipos de dispositivos, como los equipos Manos Libres.

Los mayores perjudicados con el descubrimiento y publicación de vulnerabilidades en dispositivos Bluetooth son los usuarios propietarios de los mismos, ya que las técnicas de ataque desarrolladas con el fin de explotar estos agujeros de seguridad pueden afectar a su intimidad, a la confidencialidad de sus conversaciones y a la integridad de sus dispositivos.

Es importante destacar que Bluetooth es un protocolo de comunicaciones seguro y robusto, y que las fallas de seguridad descubiertas se deben a la incorrecta implementación de los mecanismos de seguridad de Bluetooth en los dispositivos por parte de los fabricantes. La publicación de estas vulnerabilidades ha hecho reaccionar a los fabricantes de dispositivos Bluetooth, obligándoles a mejorar la seguridad en sus equipos antes de su lanzamiento en el mercado. No obstante, el estudio continúa y actualmente se siguen publicando nuevas vulnerabilidades y herramientas que permiten su explotación.

Con los continuos avances de la tecnología Bluetooth surgirán nuevos modelos de uso y se comercializarán nuevos productos, que serán nuevamente objeto de investigación por parte de grupos dedicados a la auditoría y estudio de la seguridad con el fin de descubrir nuevas vulnerabilidades que notificar a los fabricantes. En esto consiste el negocio de la seguridad informática.

Objetivos del Proyecto

El Proyecto de Fin de Carrera "Seguridad en Bluetooth" persigue los siguientes objetivos:

1. Estudio del protocolo de comunicaciones Bluetooth a un nivel más profundo del que se imparte en docencia académica.
En un futuro próximo, Bluetooth va a consolidarse como el estándar de comunicaciones de redes de área personal universal. Todos los dispositivos de nueva generación que aparecen en el mercado ya incorporan esta tecnología y su uso se está expandiendo a otros dispositivos cotidianos, como los coches o los electrodomésticos. El hecho de estudiar y conocer el funcionamiento de Bluetooth mejorará la formación tecnológica del alumno como ingeniero informático.

2. Estudio de la seguridad de un entorno innovador.
Bluetooth es una tecnología de reciente implantación y en materia de seguridad supone un terreno de investigación aún por descubrir. Puede presentar fallas de seguridad intrínsecas que los fabricantes no han tenido en cuenta a la hora de desarrollar el protocolo o implantarlo en sus dispositivos, al contrario que otros estándares de comunicaciones, donde la seguridad no plantea problemas y cada día resulta más complejo descubrir nuevas vulnerabilidades,

3. Estudio de la seguridad desde una perspectiva práctica.
Este proyecto se plantea como un estudio práctico sobre una base teórica de la seguridad en Bluetooth. Se documentarán casos prácticos y experimentos reales sobre dispositivos Bluetooth que permitirán demostrar de forma visible cómo es posible explotar vulnerabilidades con ayuda de herramientas. El hecho de llevar la teoría a la práctica y realizar pruebas con recursos reales en lugar de trabajar con simulaciones, mejorará la perspectiva del alumno a la hora de desarrollar tecnología aplicable al mundo real.

4. Dar a conocer los riesgos de Bluetooth: una necesidad ética.
La tecnología Bluetooth está cada vez más extendida y los fabricantes que implantan esta tecnología en sus dispositivos no siguen todas las recomendaciones del Bluetooth SIG, despreocupándose por la implementación de los elementos de seguridad y dejando agujeros al descubierto que pueden ser aprovechados por atacantes maliciosos para vulnerar el dispositivo comprometido. Por otro lado, el software resulta muy difícil de corregir en estos dispositivos, razón de más para asegurar la seguridad desde un punto de vista más preventivo que reactivo.

La falta de conocimiento del riesgo existente en estas fallas de seguridad para los usuarios propietarios de dispositivos Bluetooth hace que surja la necesidad ética de publicar lo sencillo que resulta atacar estos dispositivos y que de esta forma, los fabricantes tomen conciencia de lo importante que resulta cuidar el aspecto de la seguridad en la comercialización de dispositivos Bluetooth.

5. Desarrollo de herramientas de análisis de seguridad.
Toda demostración de cómo explotar vulnerabilidades en dispositivos Bluetooth incluida en este proyecto se basará en el empleo de herramientas de ataque. Algunas de estas herramientas ya han sido desarrolladas y publicadas en la red. En algunos casos, será necesario adaptar la herramienta a una plataforma de ataque específica, mientras que, en otros casos, será necesario desarrollar herramientas propias para poder llevar a la práctica las técnicas de ataque teóricas. En este proyecto se documentarán todas las herramientas utilizadas a lo largo del ciclo de vida del mismo, explicando su funcionamiento y analizando su código.

Metodología de trabajo

La planificación del proyecto será la siguiente:

Se comienza con un estudio general del estándar de comunicaciones Bluetooth, en el que se recoge la especificación, los dispositivos disponibles en el mercado, la arquitectura del protocolo, los modelos de uso, los servicios y los perfiles y, por último; los mecanismos de seguridad que ofrece Bluetooth.

Posteriormente, el estudio se centra en el aspecto concreto de la identificación y seguridad en dispositivos que funcionan con tecnología Bluetooth.

A continuación, se documentan las vulnerabilidades existentes en dispositivos Bluetooth en orden histórico de descubrimiento:

- Teléfonos móviles.
- Dispositivos Manos Libres.

Cada tipo de dispositivo contiene diferentes vulnerabilidades que pueden ser explotadas mediante diversas técnicas de ataque. Para cada una de las vulnerabilidades, se documentan los siguientes puntos:

- Descripción de la vulnerabilidad.
- Descripción teórica de la técnica de ataque.
- Desarrollo de herramientas para llevar el ataque a la práctica.
- Puesta en práctica del ataque con diferentes dispositivos vulnerables.
- Documentación de los experimentos realizados.

Capítulo

2

ESTÁNDAR DE COMUNICACIONES BLUETOOTH

2.1 – Especificación de Bluetooth

2.1.1 - Descripción del estándar

Bluetooth es la especificación que define un estándar global de comunicaciones inalámbricas para redes de área personal que permite la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia en entornos de comunicaciones móviles y estáticos.

La especificación Bluetooth está recogida por el grupo de trabajo 802.15.1 del IEEE

Los objetivos de la tecnología Bluetooth son los siguientes:

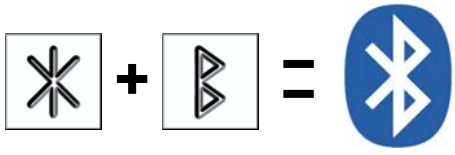
- El sistema deberá ser universal, operar en todo el mundo
- El sistema será capaz de establecer comunicación entre dos dispositivos que cumplan con las especificaciones Bluetooth, cualesquiera que sea su naturaleza: PC, teléfono móvil, accesorios de automóvil, etc.
- El emisor de radio deberá consumir poca energía, ya que debe integrarse en equipos alimentados por baterías.
- El precio del microchip transmisor deberá ser bajo (entre $4 - $5 en 2005 y con un precio objetivo de $2 - $2.50 en 2008)
- Se tratará de un sistema basado en un protocolo robusto y seguro.

2.1.2 – Etimología

El nombre *Bluetooth* procede del rey danés del siglo X llamado Harald Blatand (traducido como Harold Bluetooth), conocido por unificar las tribus en guerra de Noruega, Suecia y Dinamarca e iniciar el proceso de cristianización de la sociedad vikinga.

La elección de *Bluetooth* para denominar a esta nueva tecnología se debe a que, de la misma manera, pretende unir diferentes dispositivos, como ordenadores, teléfonos móviles, manos libres de automóvil, etc.

El logo de Bluetooth combina la representación de las runas nórdicas *Hagalaz* (transcrito por 'H') y *Berkana* (transcrito por 'B') en un mismo símbolo.



2.1.3 - Historia

En 1994, Ericsson Mobile Communications, la compañía global de telecomunicaciones con base en Suecia, comenzó un estudio de viabilidad de una interfaz de radio de baja potencia y bajo coste entre teléfonos móviles y otros accesorios, con el objetivo de eliminar los cables. El estudio era parte de un proyecto más amplio que investigaba cómo conectar diferentes dispositivos de comunicaciones a la red celular a través de un teléfono móvil. La compañía determinó que el último enlace en ese tipo de conexión debería ser un enlace de radio de corto alcance. A medida que progresaba el proyecto, se hizo evidente que este tipo de enlace de radio de corto alcance podía ser utilizado ampliamente en un gran número de aplicaciones.

El trabajo de Ericsson en esta área atrajo la atención de IBM, Intel, Nokia y Toshiba. Estas compañías decidieron formar en febrero de 1998 un grupo especial de investigación denominado SIG (Special Interest Group) Bluetooth, con el objetivo de desarrollar, promover, definir y publicar las especificaciones de esta tecnología inalámbrica de corta distancia.

En mayo del mismo año, se invitó a otras compañías a participar en el grupo: Microsoft, Lucent Technologies, 3COM y Motorola.

En julio de 1999, el grupo publicó la especificación Bluetooth 1.0, la cual constaba de dos documentos: el núcleo fundamental (core) y el perfil fundamental. El primer documento proporcionaba las especificaciones de diseño, tales como el interfaz de radio, la capa de banda base, el gestor de enlace, el protocolo de descubrimiento de servicios, el nivel de transporte y la interoperabilidad con diferentes protocolos de comunicaciones; mientras que el perfil fundamental, proporcionaba las directrices para la interoperabilidad de aplicaciones Bluetooth.

El SIG creció hasta alcanzar más de 1800 miembros en abril de 2000. Del grupo inicial de compañías promotoras, 3COM se retiró y Lucent Technologies cedió su calidad de miembro a su filial Agere Systems. De todos las compañías adjuntas al SIG, Intel es la única de todas que no ha fabricado un producto basado en Bluetooth.

Fuente: https://www.bluetooth.org/

2.1.4 - Descripción de la tecnología Bluetooth

Bluetooth incorpora las siguientes especificaciones técnicas:

- La frecuencia de radio con la que trabaja se sitúa en el rango de 2.4 a 2.48 GHz de la banda ISM (Industrial, Scientific and Medical) disponible a nivel mundial y que no requiere licencia de operador, lo que significa una compatibilidad universal entre dispositivos Bluetooth. Con el fin de evitar interferencias con otros protocolos que operen en la misma banda de frecuencias, Bluetooth emplea la técnica de salto de frecuencias (FHSS, Frequency Hopping Spread Spectrum), que consiste en dividir la banda en 79 canales (23 en España, Francia y Japón) de longitud 1 MHz y realizar 1600 saltos por segundo.

- La capacidad de transmisión vería según versiones del núcleo:
  - Versión 1.1: 723.1 Kbps
  - Versión 1.2: 1 Mbps
  - Versión 2.0 + EDR: 2.1 ~ 3 Mbps

- La potencia de transmisión se divide en 3 clases de productos:
  - Clase 1: 100 mW / 20 dBm, con un rango de ~100 m.
  - Clase 2: 2.5 mW /4 dBm, con un rango de ~10 m.
  - Clase 3: 1 mW / 0 dBm, con un rango de ~1 m.

- La tecnología Bluetooth se implementa en transceptores de corto alcance y con un precio objetivo de tan sólo 5$.

- El protocolo de banda base es una combinación de conmutación de circuitos y paquetes que la hace apropiada para voz y datos.

- Se definen dos tipos de enlaces para soportar aplicaciones de voz y datos:

  - Enlace asíncrono sin conexión (ACL, *Asynchronous Connectionless*):
    - Conexiones simétricas o asimétricas punto-multipunto entre maestro y esclavo.
    - Conexión utilizada para tráfico de datos.
    - Sin garantía de entrega, se retransmiten paquetes.
    - La máxima velocidad de envío es de 721 Kbps en una dirección 57.6 Kbps en la otra.

- Enlace síncrono orientado a conexión (SCO, *Synchronous Connection-Oriented*):
    - Conexiones simétricas punto a punto entre maestro y esclavo.
    - Conexión capaz de soportar voz en tiempo real y tráfico multimedia.
    - Velocidad de transmisión de 64 KB/s

A partir de la versión 1.0 que se ratificó en julio de 1999, se han publicado sucesivas versiones:

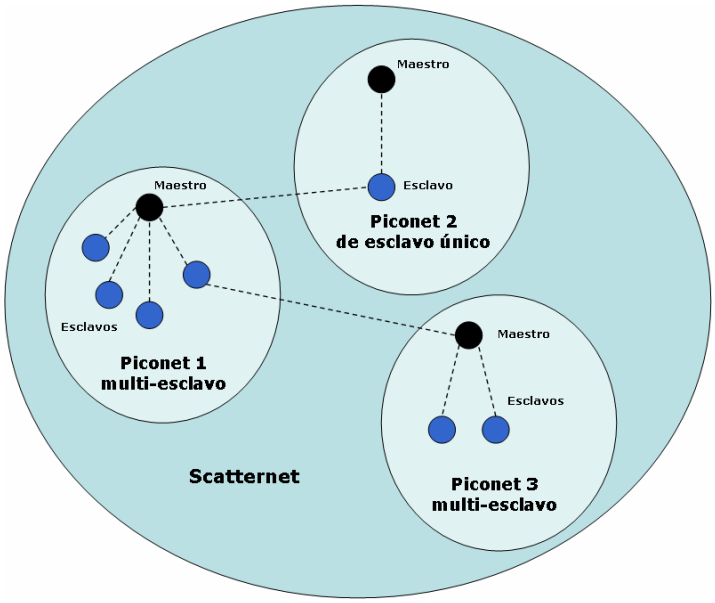
- Versión 1.1:
    - Soluciona erratas de la especificación 1.0.
    - Añade el Indicador de Calidad de Señal Recibida (RSSI)
- Versión 1.2:
    - Implementa la técnica de salto en frecuencia, *Adaptive Frequency Hopping*, para mejorar la resistencia a interferencias.
    - Introduce el tipo de enlace para aplicaciones de audio *extended Synchronous Connections* (eSCO) que mejora la calidad de voz.
    - Mejoras en el *Host Controller Interface* (HCI) para una sincronización más rápida de las comunicaciones.
- Versión 2.0:
    - Nueva versión compatible con la anterior 1.x.
    - Incorpora la tecnología *Enhanced Data Rate* (EDR), que incrementa las velocidades de transmisión hasta 3 Mbps.
    - Reducción del consumo de energía a pesar del incremento de velocidad.

2.1.5 - Topología de red Bluetooth

A diferencia de otras tecnologías LAN inalámbricas, como IEEE 802.11 (Wi-Fi), diseñadas para dispositivos que se hallen dentro o en los alrededores de un mismo edificio, los dispositivos que utilicen las redes PAN inalámbricas IEEE 802.15, incluyendo Bluetooth, podrán comunicarse en cualquier parte del mundo de forma *stand-alone*, incluso a bordo de un barco o avión y sin necesidad de utilizar equipo hardware adicional.

Cuando un dispositivo Bluetooth está dentro del radio de cobertura de otro, pueden establecer un enlace entre ellos. Hasta ocho unidades Bluetooth pueden comunicarse entre ellas y formar lo que se denomina una **Piconet** o Picorred.

La unión de varias piconets se denomina **Scatternet** o Red Dispersa.



Los dispositivos dentro de una piconet juegan dos papeles: maestro o esclavo. En todas las piconets sólo puede haber una unidad maestro, que normalmente es quien inicia la conexión, el resto de unidades Bluetooth en la piconet se denominan esclavos. Cualquier dispositivo puede realizar las funciones de maestro y esclavo, pero un mismo dispositivo únicamente puede ser maestro de una piconet.

El maestro es el dispositivo de una piconet cuyo reloj y patrón de saltos se utilizan para sincronizar a todos los demás dispositivos esclavos. Todas las unidades que participan en una piconet están sincronizadas desde el punto de vista del tiempo y de la secuencia de saltos entre canales. Cada unidad dispone de un reloj de sistema interno que determina la temporización y la secuencia de saltos que debe seguir el transceptor.

La topología Bluetooth se puede describir como una estructura de piconets múltiples. Dado que la especificación Bluetooth soporta tanto conexiones punto a punto como punto a multipunto, se pueden establecer y enlazar varias piconets en forma de scatternet. Las piconets pertenecientes a una misma scatternet no están coordinadas y los saltos de frecuencia suceden de forma independiente, es decir, todos los dispositivos que participan en la misma piconet se sincronizan con su correspondiente tiempo de reloj y patrón de saltos determinado. El resto de piconets utilizarán diferentes patrones de saltos y frecuencias de relojes distintas, lo que supone distintas velocidades de salto entre canales. Aunque no se permite la sincronización de diferentes piconets, los dispositivos pueden participar en diferentes piconets gracias a una multiplexación por división de tiempo (TDM). Esto permite a un dispositivo participar de forma secuencial en diferentes piconets, estando activo en sólo una piconet cada vez.

2.2 – Dispositivos y modelos de uso

2.2.1 - Dispositivos que incorporan tecnología Bluetooth

La tecnología Bluetooth permite la comunicación inalámbrica y el intercambio de información entre dispositivos de diversa naturaleza que cumplen las especificaciones del estándar. A continuación, se muestran dispositivos de uso cotidiano que incorporan tecnología Bluetooth organizados por categorías:

- **Audio**: Auriculares stereo, manos libres auriculares

- **Automóvil**: Sistemas integrados, manos libres, módulos GPS

- **Ordenadores Personales**: Ordenadores portátiles con Bluetooth integrado, adaptadores USB Bluetooth, servidores de acceso a otras redes

- **Periféricos**: Teclados y ratones inalámbricos, impresoras



- **Telefonía y Handhelds**: Teléfono móviles, smart phones, PDAs




- **Video e Imagen**: Cámaras de fotos, cámaras de video, proyectores


Se puede encontrar un directorio con información más detallada sobre productos que incorporan tecnología Bluetooth en la página web:

http://www.bluetooth.com/Bluetooth/Connect/Products/

2.2.2 – Escenarios y modelos de uso de Bluetooth

La posibilidad de conectar diferentes dispositivos entre sí e intercambiar voz y datos ofrece una amplia gama de escenarios y aplicaciones prácticas de Bluetooth en la vida cotidiana. A continuación se presentan una serie de modelos:

- **Intercambio de archivos e información sincronizada entre ordenadores personales**, ya sean equipos de sobremesa, ordenadores portátiles, PDAs o smart phones.

  Bluetooth permite la transferencia de archivos entre dispositivos gracias al perfil OBEX FTP. De esta forma, podemos transferir a un PC las fotografías tomadas con la cámara de un teléfono móvil, copiar las notas tomadas a mano sobre una PDA o simplemente transferir archivos de video y audio a otro equipo.

  Así mismo, también es posible sincronizar elementos tales como la agenda de contactos o el calendario de tareas con un teléfono móvil o una PDA.

- **Conexión con periféricos sin necesidad de cables.**

Bluetooth permite establecer un enlace de radiofrecuencia de corto alcance ideal para la conexión de dispositivos periféricos en un rango inferior a 10 metros. Existen multitud de periféricos que emplean tecnología Bluetooth, como teclados, ratones, impresoras, lápices digitales, módems, etc.

Asimismo, también existe una amplia gama de impresoras capaces de recibir por Bluetooth la foto a imprimir desde un teléfono móvil o una cámara digital directamente, sin necesidad de utilizar un ordenador como medio de interconexión.

Fuente: http://www.canon.com/

- **Función de Manos Libres para conversaciones telefónicas**, ya sea a través de auriculares, kits de automóvil o sistemas integrados.

  Bluetooth hace posible conversar por teléfono móvil sin necesidad de utilizar las manos para sujetar el terminal cerca del oído.

  Los auriculares Bluetooth actúan como interfaz de entrada y salida de voz y permiten libertad de movimiento con las manos, al tiempo que mantienen la confidencialidad de la llamada. Existen varios formatos disponibles, como los modelos adaptables a la oreja y las gafas de sol.

Fuente: http://www.jabra.com/

Fuente: http://www.motorola.com

Los kits de automóvil Bluetooth recogen y proyectan la voz en el interior del vehículo y permiten al conductor mantener conversaciones por teléfono sin necesidad de apartar las manos del volante.

Las marcas más prestigiosas de la industria del automóvil ya incorporan tecnología Bluetooth en sus coches, permitiendo al conductor integrar funciones del teléfono móvil con el resto de controles del vehículo. De esta forma, cuando el terminal recibe una llamada telefónica el sistema detiene la función de radio/CD y pasa a proyectar por los altavoces la conversación, asegurando que el conductor no tenga que apartar las manos del volante.

Fuente: http://www.motorola.com/

- **Sistemas de navegación GPS (Global Positioning System)**

Bluetooth ofrece un medio de comunicación inalámbrico de corto alcance ideal para el envío de coordenadas NMEA geoposicionales entre los módulos receptores GPS y los equipos visualizadores de mapas como PDAs o teléfonos móviles.

Fuente: http://www.driveblue.com/

Fuente: http://www.nokia.com

- **Marketing de proximidad por envío de publicidad**

Algunas compañías ya han comenzado campañas de publicidad en las calles basadas en el envío masivo de publicidad directa al teléfono móvil a través de Bluetooth. Emplean dispositivos emisores colocados en puntos estratégicos de elevado tránsito de personas capaces de enviar en un rango de 100 metros información personalizada que se adecua al modelo de teléfono móvil que recibe la información.

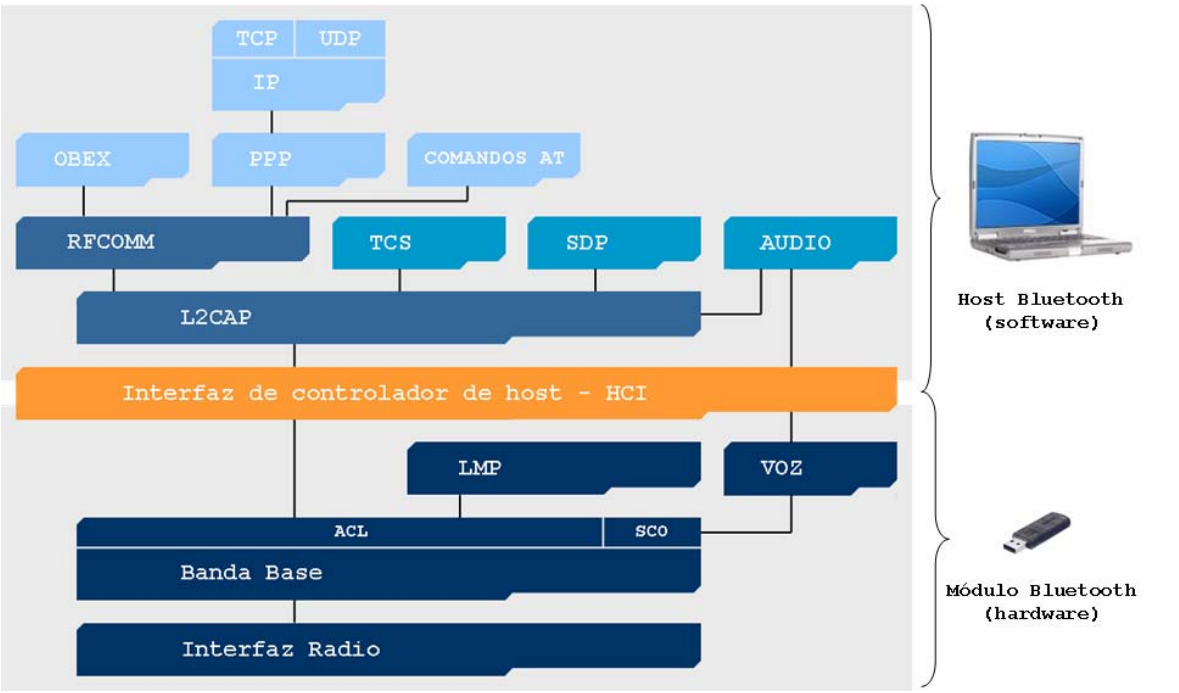
Algunos ayuntamientos han comprobado el éxito de este tipo de estrategias y han instalado sistemas de envío de información en puntos de interés general, como zonas turísticas, aeropuertos e intercambiadores de transporte público, edificios históricos y museos.

Fuente: http://www.futurlink.com/

2.3 – Arquitectura del protocolo Bluetooth

2.3.1 - La pila de protocolos Bluetooth

La pila o *stack* de protocolos Bluetooth se basa en el modelo de referencia OSI (Open System Interconnect) de ISO (Internacional Standard Organization) para interconexión de sistemas abiertos. La especificación Bluetooth utiliza una arquitectura de protocolos que divide las diversas funciones de red en un sistema de niveles. En conjunto, permiten el intercambio transparente de información entre aplicaciones diseñadas de acuerdo con dicha especificación y fomentan la interoperabilidad entre los productos de diferentes fabricantes.



La pila de protocolos Bluetooth se divide en dos zonas, cada una de las cuales se implementa en distintos procesadores:

- El **módulo Bluetooth** (hardware), encargado de las tareas relacionadas con el envío de información a través del interfaz de radiofrecuencia.

- El **host Bluetooth** (software), encargado de la parte relacionada con las capas superiores de enlace y aplicación.

Ambas zonas están comunicadas por el Interfaz de Controlador de Host (HCI).

Sobre la capa de protocolos específicos de Bluetooth, cada fabricante puede implementar su capa de protocolos de aplicación propietarios. De esta forma, la especificación abierta de Bluetooth expande enormemente el número de aplicaciones que pueden beneficiarse de las capacidades que ofrece esta tecnología inalámbrica. Sin embargo, la especificación Bluetooth exige que, a pesar de la existencia de diferentes pilas de protocolos de aplicación propietarios, se mantenga la interoperabilidad entre dispositivos que implementen diferentes pilas.

Las pilas de protocolos Bluetooth más conocidas son Widcomm, Toshiba Bluetooth Stack, Microsoft Windows XP Bluetooth y IVT BlueSoleil Stack. Linux dispone de las pilas de protocolos Bluetooth BlueZ, OpenBT y Affix, de Nokia.

2.3.2 - Capa de banda base y el interfaz de radio

En la base de la pila de protocolos Bluetooth se encuentran la capa de banda base y el interfaz de radio. Su función principal es permitir el enlace físico por radiofrecuencia (RF) entre unidades Bluetooth dentro de una picorred realizando tareas de modulación y demodulación de los datos en señales RF que se transmiten por el aire.

El nivel de banda base proporciona los dos tipos de enlace físico:

- Enlace asíncrono sin conexión (ACL, *Asynchronous Connectionless*) para tráfico de datos.

- Enlace síncrono orientado a conexión (SCO, *Synchronous Connection-Oriented*) para tráfico de audio o audio + datos.

2.3.3 - Capa de protocolo de Gestión de Enlace (LMP)

LMP (Link Manager Protocol) es el responsable de la configuración y control de enlace entre dispositivos Bluetooth. Cuando dos dispositivos Bluetooth se encuentran dentro del radio de acción del otro, el gestor de enlace (Link Manager) de cada dispositivo se comunica con su homólogo por medio de mensajes a través del protocolo LMP. Estos mensajes realizan el establecimiento del enlace entre ambos dispositivos. LMP también se encarga de las tareas relacionadas con la seguridad: autenticación y cifrado; generación, intercambio y comprobación de las claves de enlace y cifrado.

2.3.4 – Capa de Interfaz de Controlador de Host (HCI)

La capa HCI (Host Controller Interface) actúa como frontera entre las capas de protocolo relativas al hardware (módulo Bluetooth) y las relativas al software (host Bluetooth). Proporciona una interfaz de comandos para la comunicación entre el dispositivo y el firmware del módulo Bluetooth y permite disponer de una capa de acceso homogénea para todos los módulos Bluetooth de banda base, aunque sean de distintos fabricantes.

Una de las tareas más importantes del interfaz HCI es el descubrimiento de dispositivos Bluetooth que se encuentren dentro del radio de cobertura. Esta operación se denomina consulta o *inquiry* y funciona del siguiente modo:

- Inicialmente, el dispositivo origen envía paquetes *inquiry* y se mantiene en espera de recibir respuestas de otros dispositivos presentes en su zona de cobertura.

- Si los dispositivos destino están configurados en modo visible (*discoverable*) se encontrarán en estado *inquiry_scan* y en predisposición de atender estos paquetes. En este caso, al recibir un paquete *inquiry* cambiarán a estado *inquiry_response* y enviarán una respuesta al host origen con sus direcciones MAC y otros parámetros.

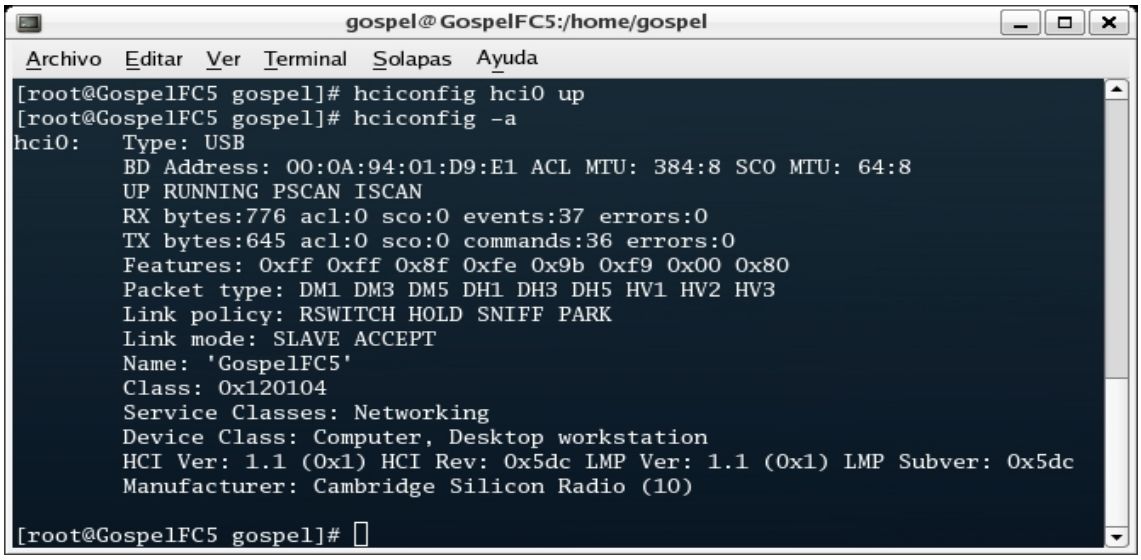
- Los dispositivos que estén configurados en modo no visible (*non discoverable*) se encontrarán en modo *inquiry_response* y, por tanto, no responderán al host origen y permanecerán ocultos.

2.3.4.1 – Direccionamiento de dispositivos Bluetooth

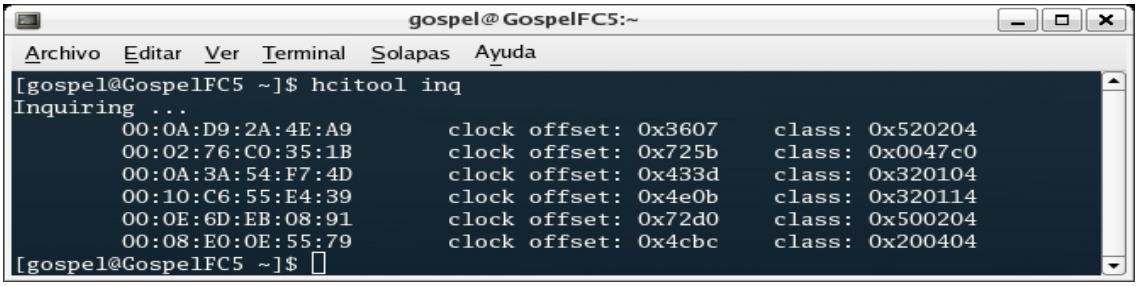
Al igual que en otros estándares de comunicaciones IEEE 802, Bluetooth utiliza direcciones MAC de 6 bytes para el direccionamiento de equipos a nivel de red. De esta forma, un dispositivo queda identificado unívocamente por su dirección MAC, comúnmente denominada *BD_ADDR*.

La pila de protocolos BlueZ para Linux dispone de dos herramientas que permiten realizar funciones específicas del interfaz HCI:

- La herramienta ***Hciconfig*** permite configurar el interfaz del módulo Bluetooth conectado al dispositivo.



- La herramienta ***Hcitool*** permite realizar operaciones relativas a la gestión de enlace con otros dispositivos Bluetooth, tales como enviar paquetes *inquiry* para la detección de equipos cercanos, resolución de nombres, identificación de clases, etc.



2.3.5 - Capa de Protocolo de Adaptación y Control del Enlace Lógico (L2CAP)

La especificación Bluetooth incluye el protocolo L2CAP (Logical Link Control and Adaptation Protocol), que se encarga de la multiplexación de protocolos, ya que el protocolo de banda base no soporta un campo *tipo* para identificar el protocolo de nivel superior al que quiere transmitir la información, por ejemplo SDP, RFCOMM y TCS.

Otra función que se realiza en el nivel L2CAP es la segmentación y recomposición de paquetes, necesaria para permitir la utilización de protocolos que utilicen paquetes de mayor tamaño que los soportados por la capa de banda base. Los paquetes L2CAP de gran tamaño se deben segmentar en múltiples paquetes de formato banda base más pequeños antes de su transmisión. En el lado del receptor, los paquetes de banda base se recomponen en paquetes L2CAP más grandes tras comprobar su integridad.
El proceso de establecimiento de la conexión L2CAP también permite el intercambio de información referente a la calidad de servicios (QoS) que se espera entre dos dispositivos Bluetooth. La implementación L2CAP en cada uno de los extremos controla los recursos utilizados por el protocolo y se asegura de que se cumplen los contratos de calidad de servicio.

La especificación L2CAP está definida únicamente para enlaces asíncronos sin conexión (ACL) y no puede existir más que un enlace entre dos dispositivos.

2.3.6 – Capa de Protocolo de Descubrimiento de Servicios (SDP)

El descubrimiento de servicios hace referencia a la capacidad de buscar y encontrar servicios disponibles en dispositivos Bluetooth. A través de los servicios, dos dispositivos pueden ejecutar aplicaciones comunes e intercambiar datos.

El protocolo SDP (Service Discovery Protocol) permite a una aplicación cliente obtener información sobre servidores SDP disponibles en otros dispositivos Bluetooth cercanos, enumerar los servicios que ofrecen y las características de dichos servicios. Después de haber localizado los servicios disponibles en un dispositivo, el usuario puede elegir aquel de ellos que resulte más apropiado para el tipo de comunicación que desea establecer.

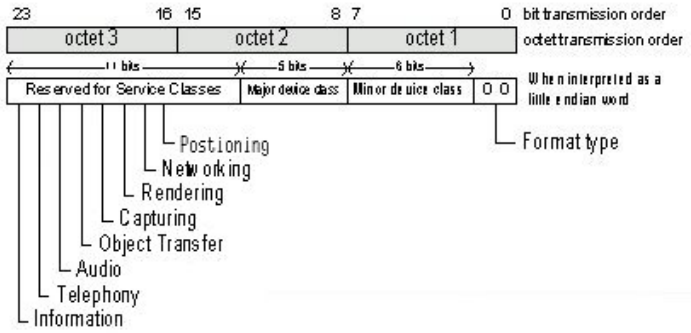
Un servicio es cualquier entidad que puede ofrecer información, ejecutar una acción o controlar un recurso. Un servicio puede estar implementado como hardware, software o una combinación de hardware y software.

2.3.6.1 – *Services Classes*

Un servicio concreto soportado por cierto dispositivo es una instancia de un *Service Class* o clase de servicio. El *Service Class* describe los servicios genéricos soportados por un dispositivo:

- Positioning (Location identification)
- Networking (LAN, Ad hoc, ...)
- Rendering (Printing, Speaker, ...)
- Capturing (Scanner, Microphone, ...)
- Object Transfer (v-Inbox, v-Folder, ...)
- Audio (Speaker, Microphone, Headset service, ...)
- Telephony (Cordless telephony, Modem, Headset service, ...)
- Information (WEB-server, WAP-server, ...)

Para dar a conocer los servicios genéricos que soporta un dispositivo Bluetooth, este incorpora en la cabecera de nivel de banda base de sus paquetes un campo Class of Device/Service que contiene información acerca de su *Service Class*.



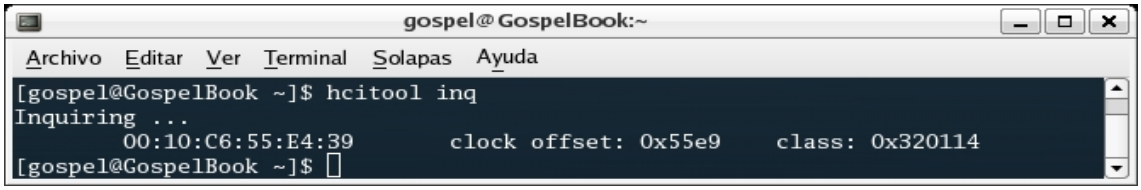
Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

El campo reservado para el *Service Class* se compone de 11 bits, del bit 23 al 13. En la especificación de banda base 1.1 de Bluetooth, se describe la siguiente relación entre los bits marcados en el campo *Service Class* y los servicios genéricos soportados por el dispositivo.

Bit no	Major Service Class
13	Limited Discoverable Mode [Ref #1]
14	(reserved)
15	(reserved)
16	Positioning (Location identification)
17	Networking (LAN, Ad hoc, ...)
18	Rendering (Printing, Speaker, ...)
19	Capturing (Scanner, Microphone, ...)
20	Object Transfer (v-Inbox, v-Folder, ...)
21	Audio (Speaker, Microphone, Headset service, ...)
22	Telephony (Cordless telephony, Modem, Headset service, ...)
23	Information (WEB-server, WAP-server, ...)

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

Conociendo el *Class of Device/Service* de un dispositivo Bluetooth, se puede averiguar fácilmente el conjunto de servicios genéricos soportados por el mismo.



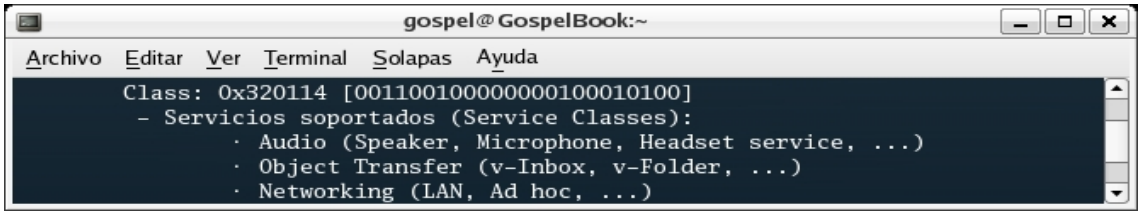
En este ejemplo, la herramienta ***Hcitool*** ha detectado un dispositivo cuyo *Class of Device/Service* es 0x320114.

Representando 0x320114 en binario, obtenemos:

Nº bit:	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Valor:	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0

Se observa que aparecen marcados los bits 21, 20 y 17, que, tal y como establece la tabla *Major Service Classes*, representan la disponibilidad de los siguientes servicios genéricos:

- bit 21: Audio (Speaker, Microphone, Headset service, ...)
- bit 20: Object Transfer (v-Inbox, v-Folder, ...)
- bit 17: Networking (LAN, Ad hoc, ...)



2.3.6.2 – *Service Record*

Toda la información relacionada con un servicio que mantiene un servidor SDP está contenida en un *Service Record* o registro individual de servicio.

Un *Service Record* consiste en una lista de atributos que describen características de un servicio: Service Name, Service Description, Provider Name, Service Record Handle, Service Class ID List, Service Record State, Service ID, Protocol Description List, Browse Group List, Language Base Attribute ID List, Service Info Time To Live, Service Avaliability y Bluetooth Profile Descriptor List.

El protocolo SDP permite realizar dos tipos de operaciones relacionadas con el descubrimiento de servicios en dispositivos Bluetooth: búsqueda y enumeración de servicios.

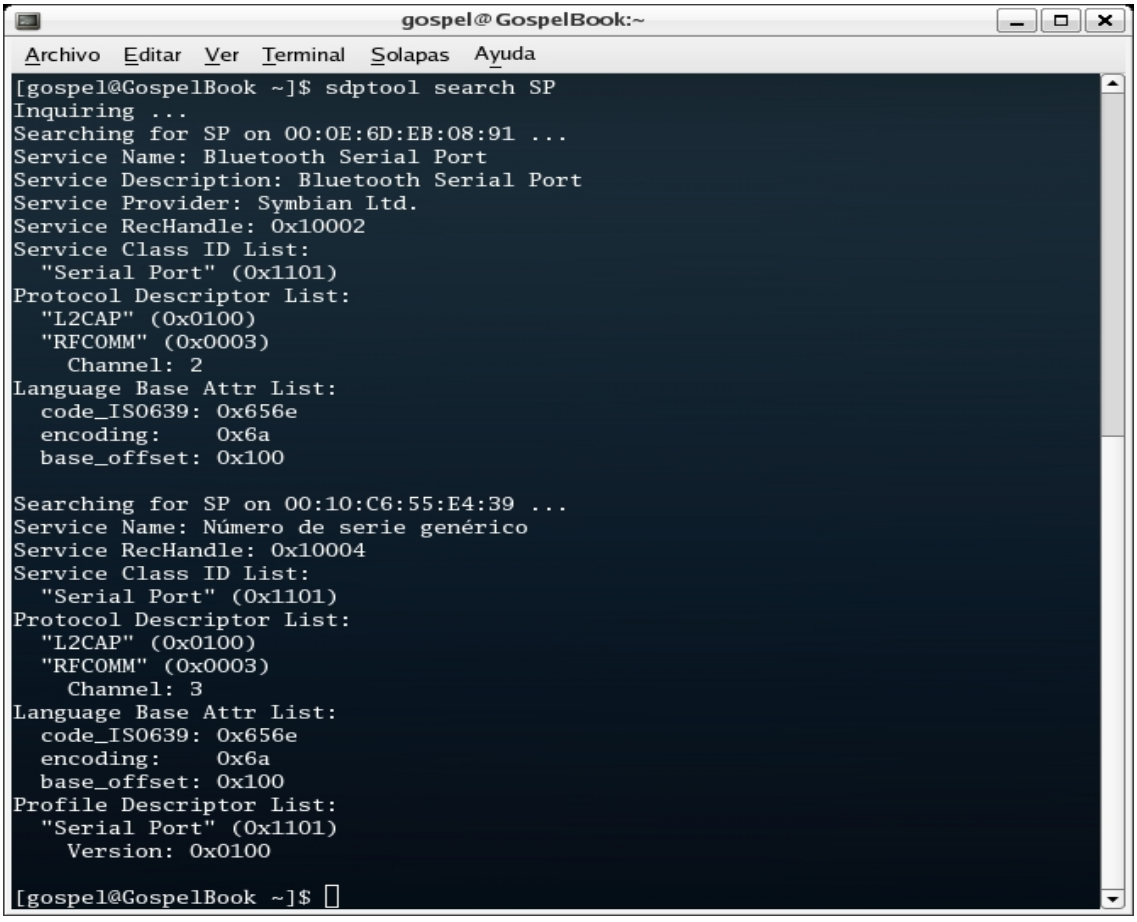
- La operación búsqueda de servicios (*Service Searching*) permite a un cliente SDP encontrar dispositivos que ofrecen un servicio específico.

- La operación enumeración de servicios (*Service Browsin*) permite a un cliente SDP conocer los servicios ofrecidos por un determinado dispositivo.

En ambos casos, el resultado de la petición SDP devolverá al cliente que la originó una lista de servicios descubiertos acompañada por la definición de los mismos a través de sus *Service Records*.

La pila de protocolos BlueZ para Linux dispone de una herramienta de gestión SDP llamada ***Sdptool*** que permite llevar a cabo los dos tipos de operaciones anteriormente descritos.

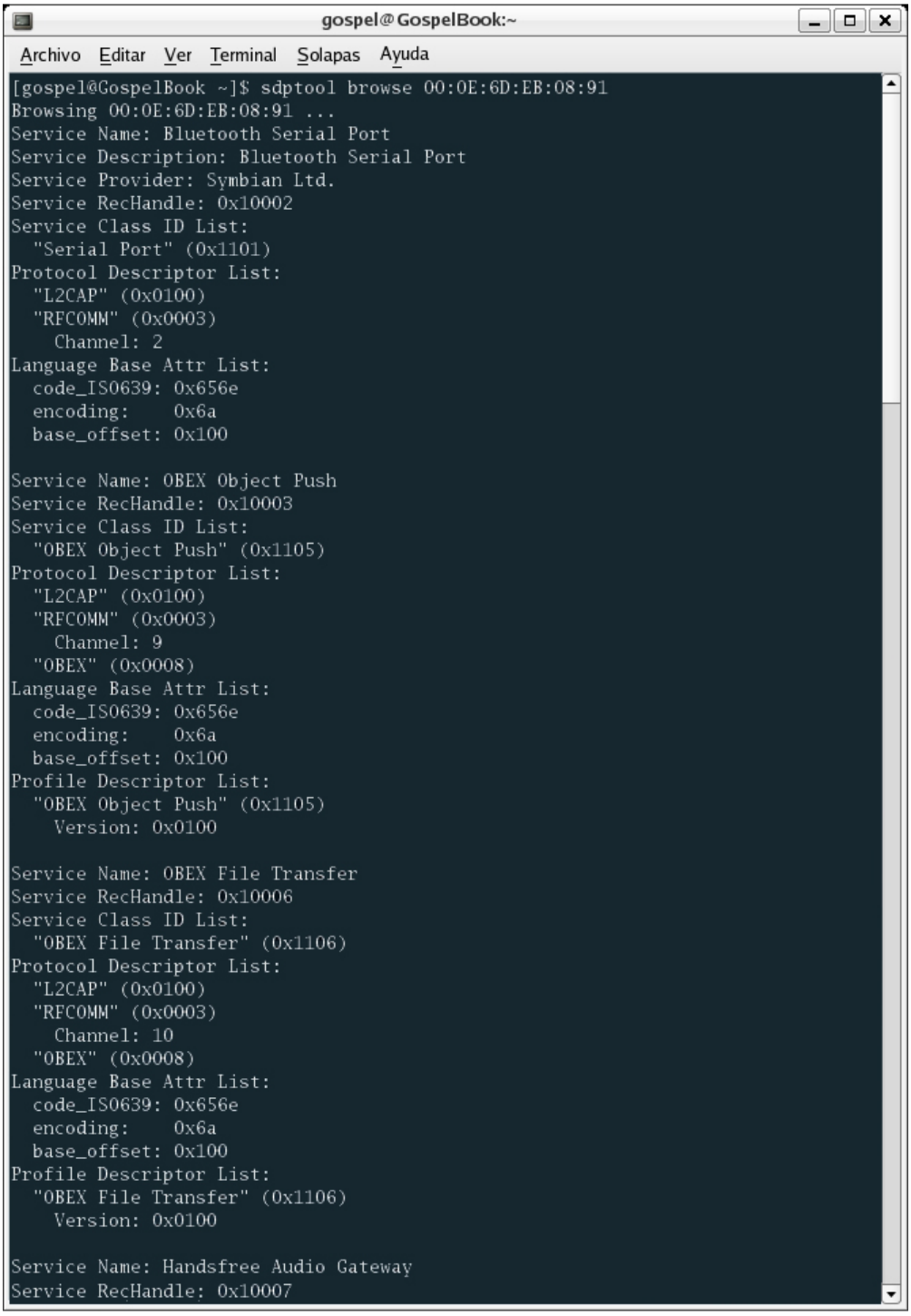
Permite buscar dispositivos cercanos que ofrezcan un servicio específico, como por ejemplo Puerto Serie, Manos Libres, etc.



La especificación de la herramienta *sdptool* establece los siguientes servicios disponibles para operaciones de búsqueda:

```
DID, SP, DUN, LAN, FAX, OPUSH, FTP, HS, HF, SAP, NAP, GN, PANU, HID, CIP, CTP, A2SRC, A2SNK, AVRCT, AVRTG, SR1, SYNCML, ACTIVESYNC, HOTSYNC, PALMOS, NOKIA PCSUITE.
```

Sdptool también permite enumerar todos los servicios disponibles en un determinado dispositivo.



2.3.7 – Capa RFCOMM

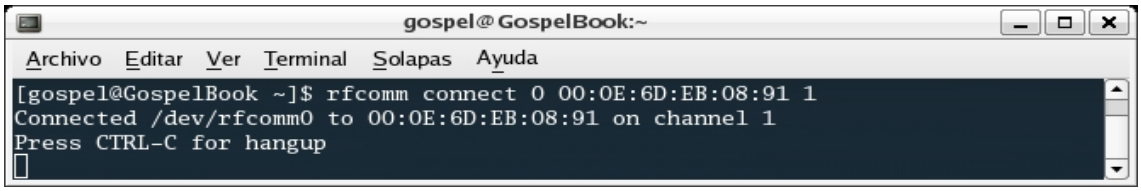
El protocolo RFCOMM (Radio Frequency Communication) es un protocolo de emulación de línea serie basado en el estándar ETSI TS 07.10. Proporciona una emulación de los puertos serie RS-232 sobre el protocolo L2CAP.

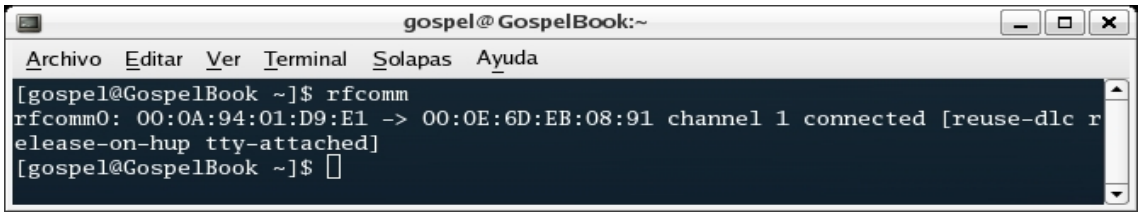
Este protocolo de "sustitución de cable serie" emula las señales de control y datos RS-232 sobre la banda base, proporcionando capacidades de transporte a los servicios de niveles superiores que utilizan el cable serie como mecanismo de transporte.

Para los propósitos de RFCOMM, un camino de comunicación directa involucra siempre a dos aplicaciones que se ejecutan en dos dispositivos distintos extremos de la comunicación. Entre ellos existe un segmento que los comunica, en este caso, un enlace Bluetooth desde un dispositivo al otro. RFCOMM pretende soportar aquellas aplicaciones que utilizan los puertos serie de los dispositivos donde se ejecutan.

RFCOMM es un protocolo de transporte sencillo que soporta hasta 9 puertos serie RS-232 y permite hasta 60 conexiones simultáneas (canales RFCOMM) entre dos dispositivos Bluetooth.

El comando ***Rfcomm*** de la pila de protocolos BlueZ para Linux permite establecer comunicación con un dispositivo Bluetooth.





2.3.8 – Protocolo OBEX

OBEX (OBject EXchange) es un protocolo de nivel de sesión desarrollado originalmente por la asociación IrDA (Infrared Data Association) con el nombre de IrOBEX. Su objetivo es soportar el intercambio de objetos de forma simple y espontánea. OBEX se basa en el modelo cliente/servidor y es independiente del mecanismo de transporte, aunque en la implementación de OBEX en la especificación Bluetooth sólo se utiliza RFCOMM como nivel de transporte.

2.3.9 – Protocolos adoptados PPP

La especificación Bluetooth emplea varios protocolos existentes que se reutilizan para diferentes propósitos en los niveles superiores. El objetivo de la implementación de estos protocolos es permitir que aplicaciones antiguas funcionen con la tecnología inalámbrica Bluetooth y ayudar a asegurar un correcto funcionamiento e interoperabilidad de esas aplicaciones con aplicaciones modernas diseñadas específicamente para dispositivos Bluetooth.

Bluetooth utiliza el protocolo PPP desarrollado por el IETF (Internet Engineering Task Force), que define cómo se transmiten los datagramas IP sobre enlaces punto-a-punto, para garantizar la interoperabilidad de dispositivos Bluetooth con aplicaciones basadas en protocolos TCP y UDP en última instancia.

2.3.10 – Comandos AT

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal módem. Los comandos AT se denominan así por la abreviatura de *attention*.

El juego de comandos AT fue desarrollado en 1977 por Dennis Hayes como un interfaz de comunicación con un módem para poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono. Más adelante, fueron las compañías Microcomm y US Robotics las que siguieron desarrollando y expandiendo el juego de comandos hasta universalizarlo.

La telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales.

El juego de comandos AT puede encontrarse en la documentación técnica de los terminales GSM y permite acciones tales como realizar llamadas de datos o de voz, leer y escribir entradas en la agenda de contactos y gestión de mensajes SMS, además de muchas otras opciones de configuración del terminal.

Se puede encontrar un resumen de los comandos AT GSM más importantes en el **Anexo I – Juego de comandos AT GSM**.

La implementación de los comandos AT es específica del terminal GSM y no depende del canal de comunicación a través del cual los comandos sean enviados, ya sea cable de serie, canal Infrarrojos o Bluetooth. De esta forma, existen en el mercado distintos teléfonos móviles que implementan el juego completo de comandos AT o sólo parcialmente, en cuyo caso implementarán uno o varios de los siguientes bloques de comandos AT:

- Bloque básico de comandos AT, relacionado con la configuración y el envío de instrucciones al terminal. Permite llevar a cabo, entre otras, las siguientes operaciones: Realizar llamadas de voz y de datos, configurar desvíos de llamadas, obtener información básica sobre la marca, modelo e IMEI (Internacional Mobile Equipment Identity) del terminal, así como del nivel de batería, calidad de cobertura, etc.

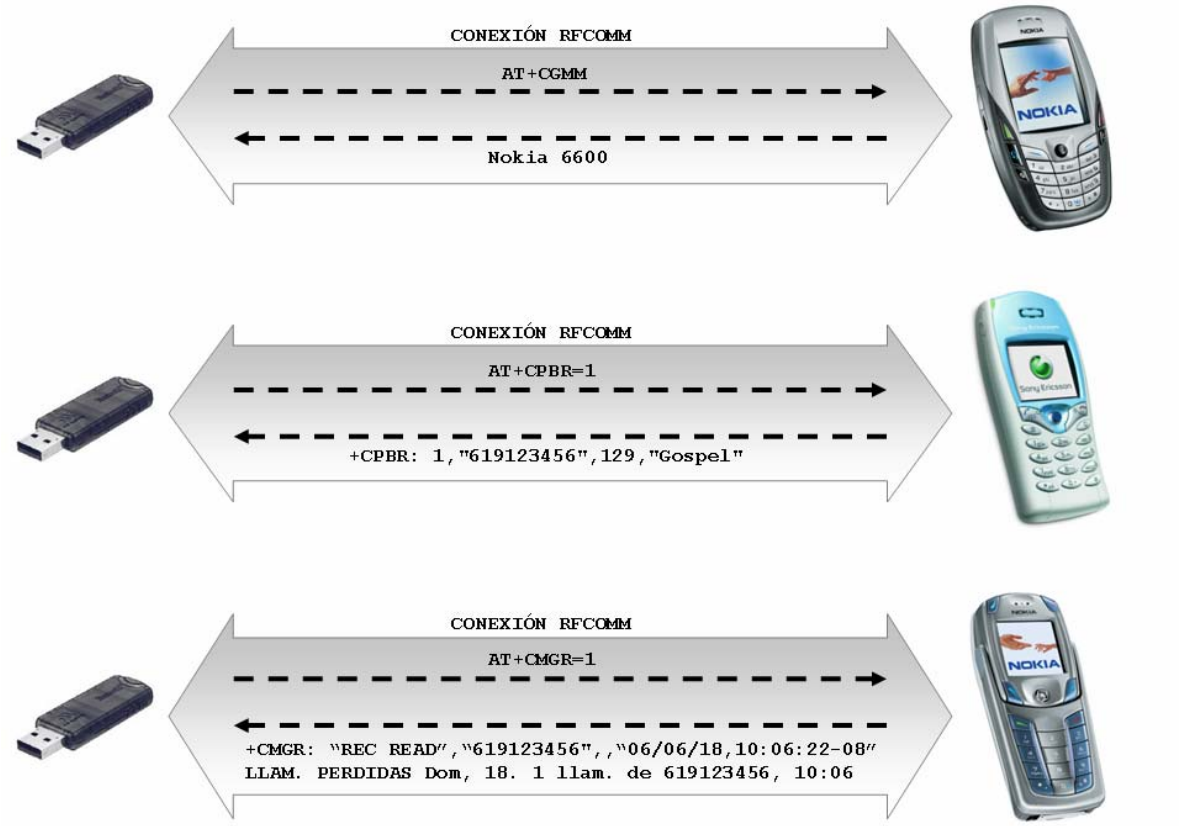
- Bloque de comandos AT referido a la gestión de la agenda de contactos, ya sea la memoria contenida en la tarjeta SIM o la lista de últimas llamadas realizadas, perdidas y recibidas almacenada en el terminal. Se pueden llevar a cabo las siguientes acciones: leer un contacto, añadir y eliminar una entrada de la agenda, buscar un contacto por nombre, etc.

- Bloque de comandos AT referido a la gestión de mensajes SMS. Permite ejecutar las siguientes operaciones: obtener el listado de los mensajes SMS almacenados en memoria, leer un mensaje SMS de la bandeja de entrada, eliminar un mensaje SMS existente, escribir un nuevo mensaje SMS, enviar mensajes SMS, etc.

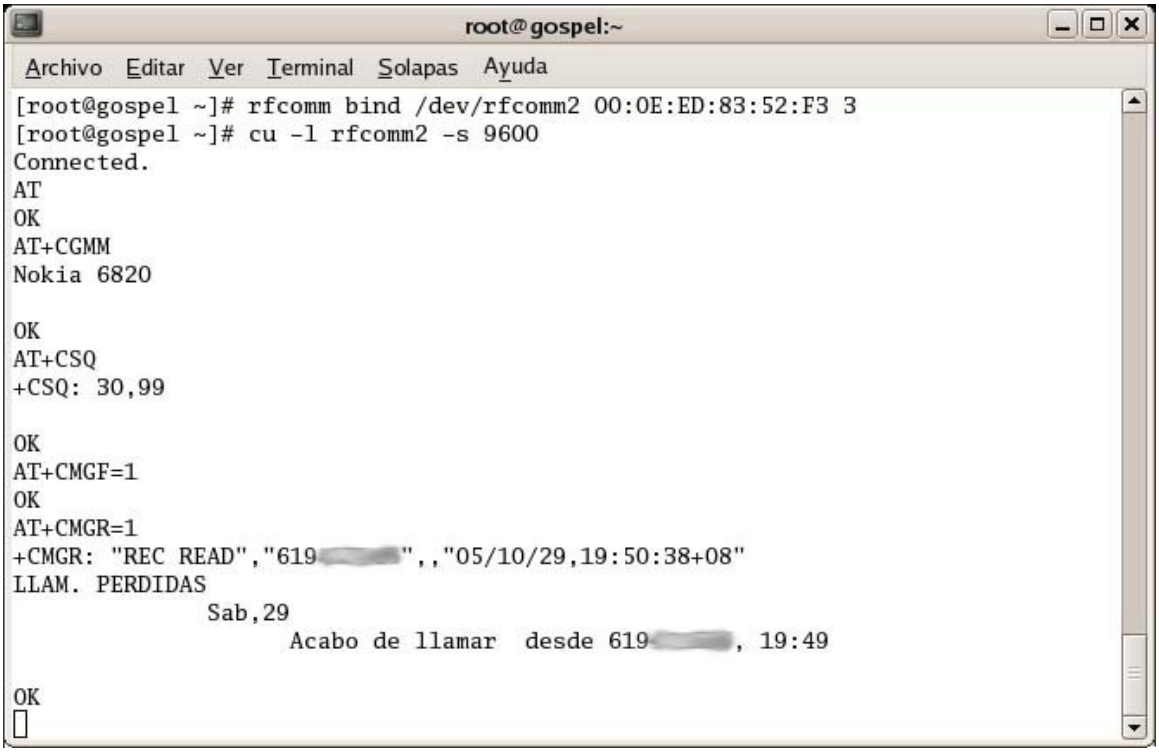
Ejemplo de tabla comparativa con el grado de implementación de los bloques de comandos AT en algunos de los teléfonos móviles más populares:

Modelo	Soporta Bloque básico	Bloque Agenda	Bloque SMS
Nokia™ 6820	✓		✓
Sony-Ericsson™ T68i	✓	✓	
Nokia™ 6600	✓		

Esquema gráfico del modelo petición/respuesta de los comandos AT:



Ejemplo de una sesión de comandos AT con un teléfono móvil:



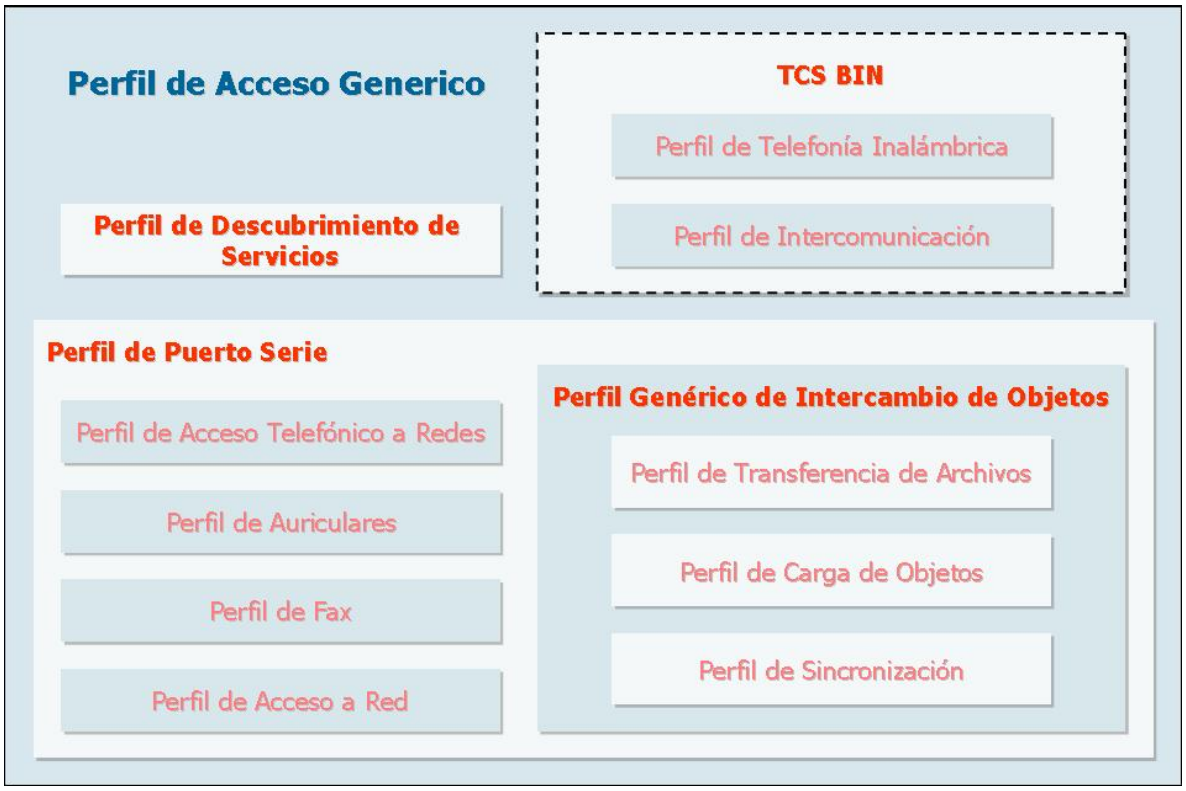
2.4 – Perfiles Bluetooth

2.4.1 – Perfiles genéricos de Bluetooth

El SIG Bluetooth ha identificado varios modelos de uso del estándar de comunicaciones Bluetooth, cada uno de los cuales está acompañado por un *perfil*. Los perfiles definen los protocolos y características que soportan un modelo de uso particular. Esto garantiza la interoperabilidad, ya que si dos dispositivos de distintos fabricantes cumplen con la misma especificación del perfil Bluetooth, podemos esperar que interactúen correctamente cuando se utilicen para un uso particular.

Un perfil define los mensajes específicos y procedimientos usados para implementar una característica. Algunas características son obligatorias y algunas pueden ser opcionales.

Se definen cuatro perfiles genéricos que contienen la especificación de los perfiles específicos: el Perfil de Acceso Genérico (GAP, Generis Access Profile), el Perfil de Puerto Serie (SPP, Serial Port Profile), el Perfil de Aplicación de Descubrimiento de Servicios (SDAP, Service Discovery Application Profile) y el Perfil Genérico de Intercambio de Objetos (GOEP, Generic Object Exchange Profile).



2.4.1.1 – Perfil de Acceso Genérico

El Perfil de Acceso Genérico (GAP, Generis Access Profile) define los procedimientos generales para descubrir dispositivos Bluetooth, así como los procedimientos de gestión de enlace para establecer una conexión entre dos dispositivos Bluetooth.

El Perfil GAP debe implementarse en cualquier dispositivo Bluetooth para asegurar la interoperabilidad básica y la coexistencia con otros dispositivos, independientemente del tipo de aplicación que soporten. Los dispositivos que además cumplan otro perfil Bluetooth pueden emplear adaptaciones de los procedimientos genéricos, tal como se especifiquen en ese perfil. Sin embargo, deben seguir siendo compatibles con el perfil GAP en el nivel de procedimientos genéricos.

2.4.1.2 – Perfil de Puerto Serie

Cuando la tecnología inalámbrica Bluetooth se utiliza para sustituir al cable, se emplea el Perfil de Puerto Serie (SPP, Serial Port Profile) para el canal resultante orientado a conexión. Este perfil está construido sobre el Perfil de Acceso Genérico y define cómo deben configurarse los dispositivos Bluetooth para emular una conexión a través de un cable serie utilizando RFCOMM, un protocolo de transporte sencillo que emula los puertos serie RS-232 entre dispositivos homólogos.

Las aplicaciones ejecutadas en los dispositivos son normalmente aplicaciones heredadas que esperan que la comunicación tenga lugar a través de un cable serie. Cualquier aplicación heredada puede ser ejecutada sobre cualquiera de los dos dispositivos utilizando el puerto serie virtual como si los conectara un cable físico, con señalización de control RS-232; pudiendo necesitar la ayuda, en algunos casos, de una aplicación auxiliar que utilice la especificación Bluetooth a ambos lados del enlace.

2.4.1.3 – Perfil de Aplicación de Descubrimiento de Servicios

El Perfil de Aplicación de Descubrimiento de Servicios (SDAP, Service Discovery Application Profile) describe las características y procedimientos utilizados para descubrir servicios registrados en otros dispositivos Bluetooth y obtener información acerca de esos servicios.

El Perfil SDAP utiliza el Protocolo de Descubrimiento de Servicios SDP, incluido en la pila de protocolos Bluetooth, para localizar los servicios disponibles en dispositivos situados dentro del radio de acción de un dispositivo Bluetooth. El procedimiento de descubrimiento de servicios en dispositivos próximos no es automático, se requiere que el usuario invoque específicamente al protocolo SDP mediante la Aplicación de Descubrimiento de Servicios. Una vez que se crea el enlace con un dispositivo determinado, se pueden localizar los servicios que ofrece y estos pueden ser seleccionados a través del interfaz de usuario según el tipo de aplicación que se desee ejecutar.

El protocolo SDP permite realizar dos tipos de operaciones relacionadas con el descubrimiento de servicios en dispositivos Bluetooth:

- Búsqueda de servicios (*Service Searching*): permite localizar dispositivos cercanos que ofrezcan un servicio específico.

- Enumeración de servicios (*Service Browsing*): permite conocer los servicios ofrecidos por un determinado dispositivo.

2.4.1.4 – Perfil Genérico de Intercambio de Objetos

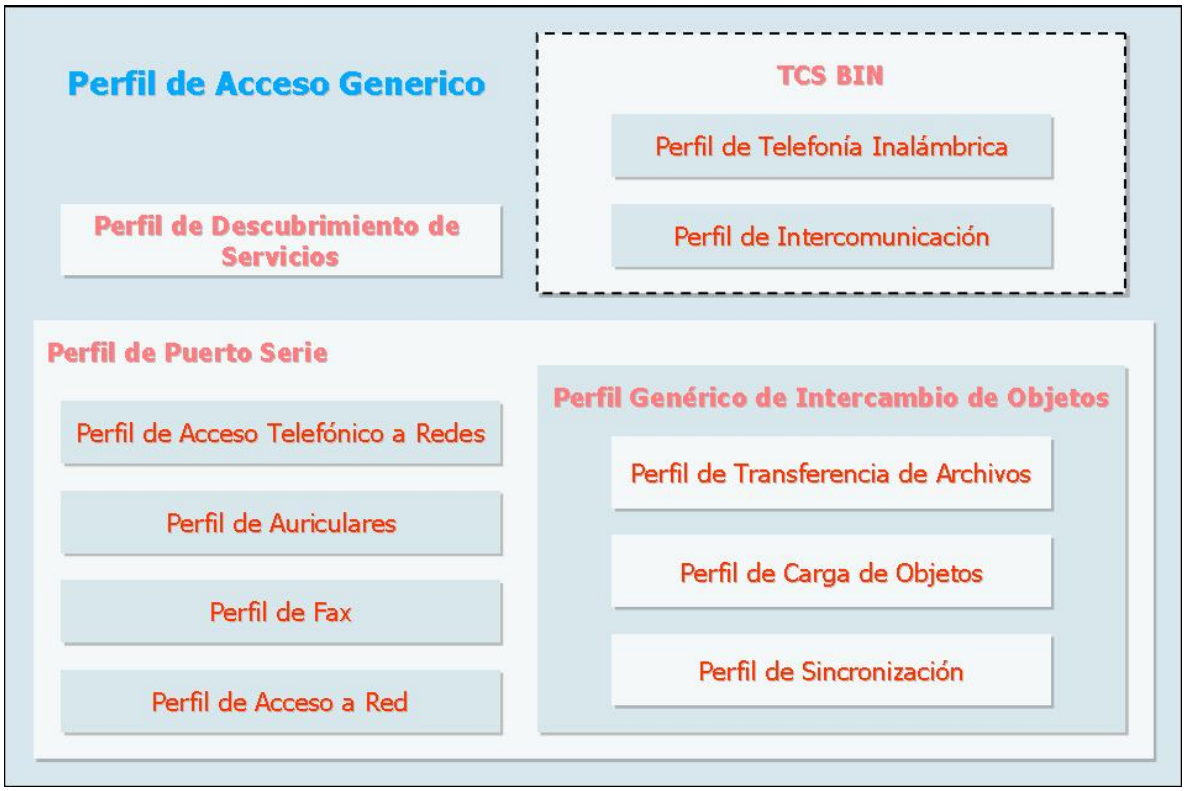
El Perfil Genérico de Intercambio de Objetos (GOEP, Generic Object Exchange Profile) define cómo deben soportar los dispositivos Bluetooth los modelos de uso de intercambio de objetos. Incluye tres perfiles asociados a modelos de uso específicos basados en el protocolo OBEX (OBject EXchange): el Perfil de Transferencia de Archivos (OBEX File Transfer), el Perfil de carga de objetos (OBEX Object Push) y el Perfil de Sincronización.

Como se describirá más adelante, OBEX permite escenarios de conexión rápida: transferencia-desconexión (OBEX Object Push) y también permite el establecimiento de sesiones en las que las transferencias tienen lugar durante un período de tiempo, manteniendo la conexión incluso cuando esté inactiva (OBEX File Transfer).

El uso principal de OBEX se realiza en aplicaciones de *carga* y *descarga* de archivos. Se basa en el modelo cliente/servidor. Bajo el Perfil Genérico de Intercambio de Objetos, un cliente *carga* o envía objetos de datos en un servidor mediante la operación PUT del protocolo OBEX; o bien *descarga* o recibe objetos de datos desde un servidor mediante la operación GET del protocolo OBEX.

2.4.2 – Perfiles Bluetooth para modelos de uso

Se han identificado cuatro perfiles genéricos (GAP, SPP, SDAP y GOEP), sobre los que se definen los diferentes perfiles específicos para modelos de uso. Estos perfiles Bluetooth para modelos de uso son múltiples y variados, y se implementan de manera opcional e independiente por cada fabricante y tipo de dispositivo.



La especificación Bluetooth 1.0 define los siguientes perfiles:

- Perfil de Telefonía Inalámbrica (CTP, Cordless Telephony Profile)
- Perfil de Intercomunicación (IP, Intercom Profile)
- Perfil de Puerto Serie (SP, Serial Port Profile)
- Perfil de Acceso Telefónico a Redes (DUN, Dial-Up Networking)
- Perfil de Auriculares (HS, HeadSet Profile)
- Perfil de Fax (FP, Fax Profile)
- Perfil de Acceso a Red (LAP, LAN Access Profile)
- Perfil de Transferencia de Archivos (FTP, File Transfer Profile)
- Perfil de Carga de Objetos (OPUSH u OPP, Object Push Profile)
- Perfil de Sincronización (Sync, Synchronization Profile)

Adicionalmente, los siguientes perfiles han sido recientemente aprobados por el SIG o están en fase de desarrollo:

- ESDP, Extended Service Discovery Profile
- A2DP, Advanced Audio Distribution Profile
- AVRCP, Audio Video Remote Control Profile
- BIP, Basic Imaging Profile
- BPP, Basic Printing Profile
- CIP, Common ISDN Access Profile
- GAVDP, Generic Audio Video Distribution Profile
- HFR, Hands-Free Profile
- HCRP, Hardcopy Cable Replacement Profile
- HID, Human Interface Device Profile
- PAN, Personal Area Networking Profile
- SAP, SIM Access Profile

2.4.2.1 – Perfil de Acceso Telefónico a Redes

El Perfil de Acceso Telefónico a Redes (DUN, Dial-Up Networking) define los protocolos y procedimientos utilizados por dispositivos tales como módems y teléfonos móviles para implementar el modelo de uso denominado *puente hacia Internet*. El escenario posible más habitual para este modelo es el uso del teléfono móvil como módem inalámbrico para conectar un PC a un servicio de acceso telefónico a Internet.

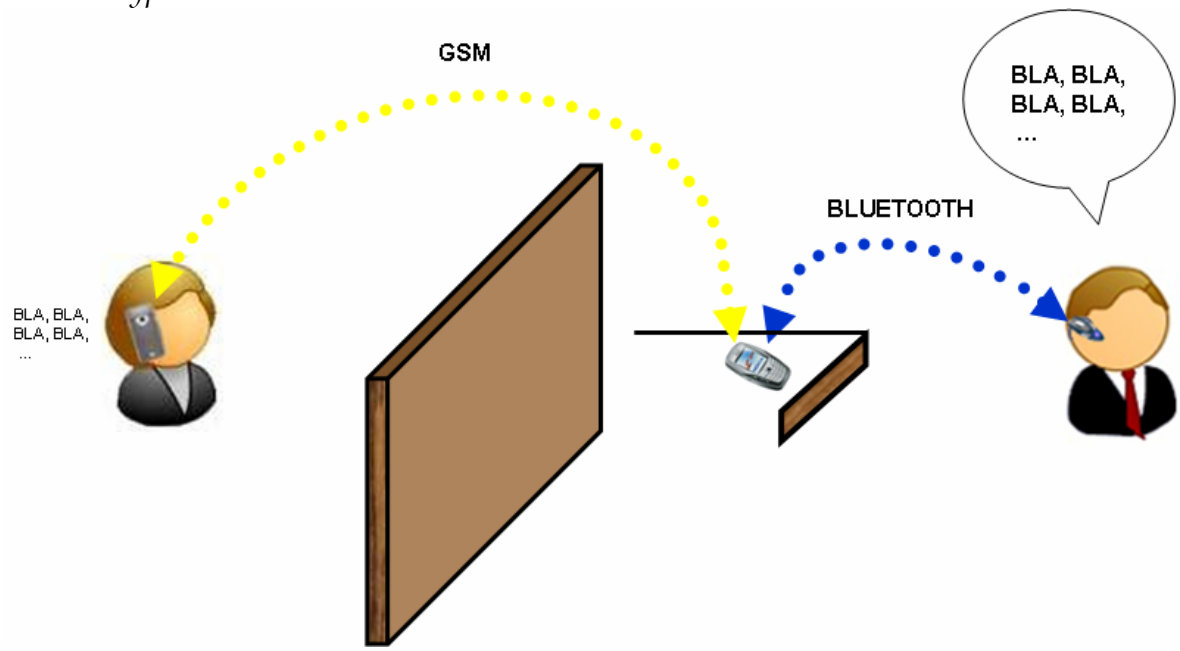
2.4.2.2 – Perfil de Auriculares

El Perfil de Auriculares (HS, HeadSet Profile) define los protocolos y procedimientos para el modelo de uso que permite utilizar un dispositivo auricular de última generación como interfaz de entrada y salida de audio de otro dispositivo, generalmente un teléfono móvil o un PC, con el propósito de incrementar la libertad de movimiento del usuario al mismo tiempo que se mantiene la confidencialidad de la conversación.

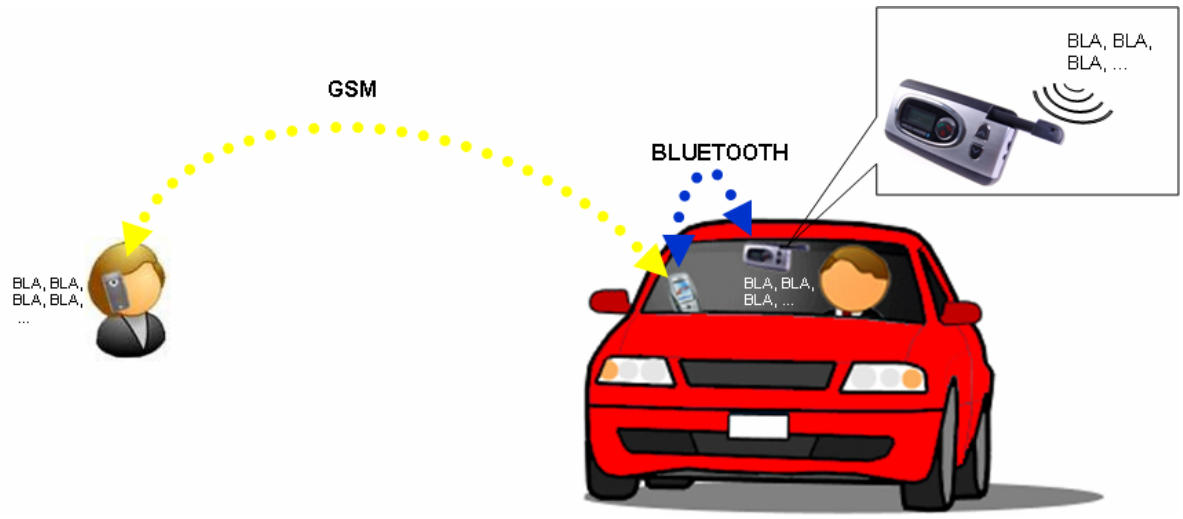
Se definen dos roles para los dispositivos que implementan el Perfil de Auriculares: pasarela de audio y auricular. El dispositivo *pasarela de audio* es aquel que inicia el procedimiento de conexión, mientras que el dispositivo *auricular* se define como el que actúa como mecanismo de entrada y salida de audio remotas para la pasarela de audio.

El modelo de uso del Perfil de Auriculares permite multitud de configuraciones y define tres escenarios de uso habituales:

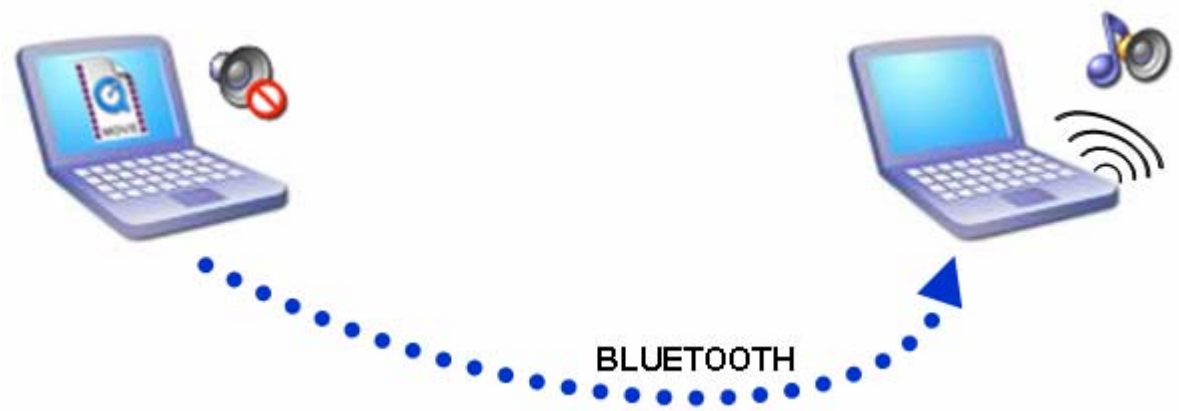
- **Manos Libres Auriculares** (Hands-Free HeadSet) conectado a un teléfono móvil: Permite al usuario mantener conversaciones telefónicas sin necesidad de acercar el terminal al oído. Su empleo puede extenderse a comunicaciones con PCs, para aplicaciones de VoIP (Voz sobre IP) como *Skype*.



- **Manos Libres de automóvil** (Hands-Free Car Kit) conectado a un teléfono móvil: Permite al usuario mantener conversaciones telefónicas en el interior de un vehículo sin necesidad de apartar las manos del volante para sostener el teléfono móvil.



- **Pasarela de audio** entre dos dispositivos Bluetooth cualesquiera: Permite a un usuario configurar dos equipos Bluetooth, que no tienen porqué tratarse de auriculares, sino simples PCs o PDAs, y establecer una pasarela de audio entre los dos, de forma que el audio que reproduce el software de un dispositivo, se transmite al otro dispositivo a través del enlace SCO (Synchronous Connection Oriented) y puede ser proyectado por los altavoces del segundo. Así mismo, el audio recogido por el micrófono de un dispositivo se transmite al otro dispositivo, donde puede ser grabado en un archivo de sonido.



establecimiento y la liberación del enlace SCO. El auricular conecta y desconecta directamente los flujos internos de audio durante el establecimiento y liberación del enlace SCO. Una vez que el enlace está establecido, existe una transferencia válida de audio sobre el enlace SCO en ambas direcciones.

2.4.2.3 – Perfil de Fax

El Perfil de Fax (FP, Fax Profile) define los protocolos y procedimientos utilizados por aquellos dispositivos que implementen la parte de fax del modelo de uso llamado *punto de acceso a datos en redes WAN*. Un teléfono móvil o un módem que utilice tecnología Bluetooth puede ser utilizado por un PC como dispositivo *fax* inalámbrico para enviar y recibir mensajes de fax.

2.4.2.4 – Perfil de Acceso a Red

El Perfil de Acceso a Red (LAP, LAN Access Profile) define cómo los dispositivos Bluetooth pueden acceder a los servicios de una LAN (Local Area Network) utilizando el protocolo PPP sobre RFCOMM, y cómo puede utilizarse el mismo protocolo PPP para conectar en red dos dispositivos utilizando Bluetooth. En este modelo de uso, varios terminales de datos utilizan un punto de acceso a la red (LAP, LAN Access Point) como conexión inalámbrica a una red de área local, de forma que operan como si estuviesen conectados a la red directamente.

PPP (Point to Point Protocol) es un estándar de la IETF utilizado ampliamente como medio de acceso a redes. Aunque PPP es capaz de soportar varios protocolos de red (IP, IPX, etc.), el Perfil de Acceso a Red no obliga al uso de ningún protocolo en particular. El Perfil de Acceso a Red simplemente define cómo se soporta PPP para proporcionar acceso a la LAN a uno o múltiples dispositivos Bluetooth y para establecer una comunicación PC a PC utilizando conexiones PPP sobre una emulación de cable serie a través de RFCOMM.

2.4.2.5 – Perfil de Transferencia de Archivos

El Perfil de Transferencia de Archivos (FTP, File Transfer Profile) soporta el modelo de uso de *transferencia de archivo*s a través del protocolo OBEX File Transfer, el cual ofrece la capacidad de transferir objetos de datos (archivos y carpetas) de un dispositivo Bluetooth a otro, así como navegar por los contenidos de las carpetas del dispositivo remoto.

Los dispositivos que implementan el Perfil de Transferencia de Archivos pueden actuar como cliente o como servidor. El dispositivo *cliente* es aquel que inicia la operación de envío o extracción de objetos al y desde el dispositivo *servidor*. El *servidor* es el dispositivo Bluetooth remoto que proporciona un servidor de intercambio de objetos a través de los comandos OBEX. Los servidores pueden imponer políticas de restricción de permisos de lectura y escritura, para evitar la creación y borrado de carpetas y archivos.

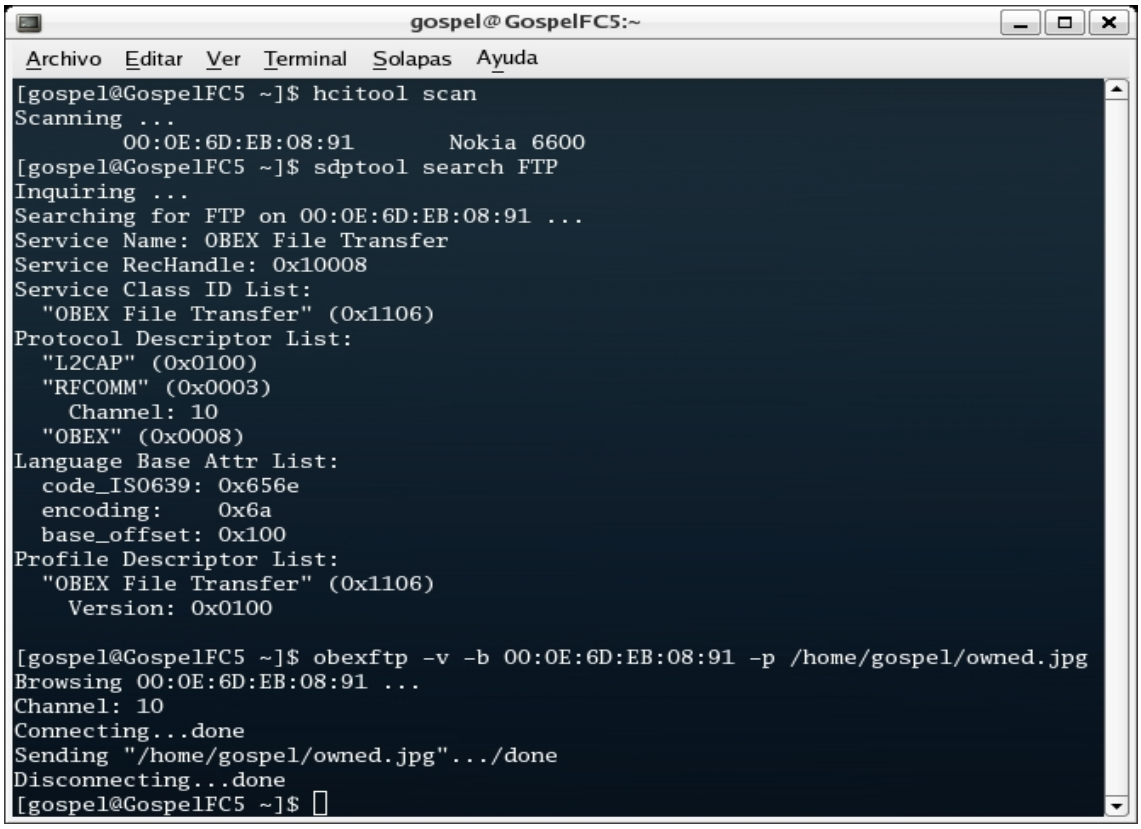
Se definen las siguientes operaciones en el Perfil de Transferencia de Archivos:

- Navegar por la jerarquía de carpetas.
- Listar el contenido de una carpeta.
- Extraer objetos, mediante el comando GET.
- Enviar objetos, mediante el comando PUT.
- Borrar objetos.

A continuación se muestra un ejemplo práctico de una aplicación basada en el protocolo OBEX File Transfer, ***Obexftp***.

Obexftp es una herramienta *open source* basada en el proyecto ***OpenObex*** (http://openobex.triq.net/), que pretende implementar el protocolo OBEX en plataformas Linux, para lo cual utiliza la pila de protocolos BlueZ. **Obexftp** permite la transferencia de archivos hacia o desde cualquier dispositivo que soporte el protocolo OBEX File Transfer.

En la siguiente captura, se muestra ***Obexftp*** en acción:



El procedimiento de transferencia de archivos con ***Obexftp*** transcurre de la siguiente forma. En primer lugar, se localiza el dispositivo Bluetooth al cual se quiere transferir el archivo con ayuda del comando ***hcitool scan***, luego se comprueba que soporta el Perfil de Transferencia de Archivos (FTP) con ayuda del comando ***sdptool search FTP*** y finalmente se envía el archivo a través de la herramienta ***Obexftp***.

2.4.2.6 – Perfil de Carga de Objetos

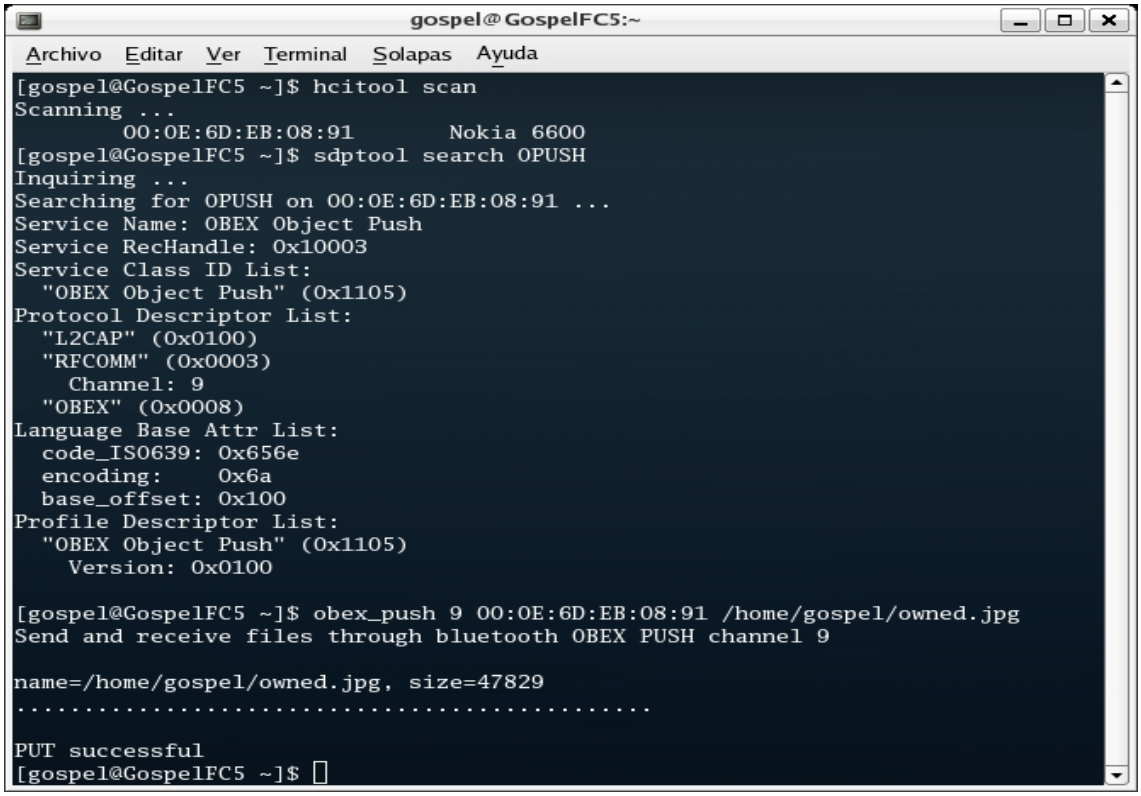
El Perfil de Carga de Objetos (OPUSH u OPP, Object Push Profile) define los requisitos de aplicación para implementar el modelo de uso de *carga de objetos* a través del protocolo OBEX Object Push, el cual ofrece la capacidad de cargar y descargar objetos de datos de un dispositivo Bluetooth a otro.

Inicialmente, el Perfil de Carga de Objetos se utilizaba para cargar y descargar objetos tales como citas en formato *vCalendar* o tarjetas de visita en formato *vCard* de otro dispositivo, lo que permitía el intercambio de tarjetas de visita entre dos dispositivos Bluetooth. Actualmente, el perfil conserva esta funcionalidad, aunque también se utiliza para transferencia rápida de archivos.

A continuación se muestran dos ejemplos prácticos de aplicaciones basadas en el protocolo OBEX Object Push, ***ObexPush*** y ***Ussp-push***.

2.4.2.6.1 - ObexPush

ObexPush es una utilidad incluida en el paquete *openobex-apps* (dependiente de cada distribución Linux) que contiene aplicaciones basadas en el proyecto **OpenObex**, que implementa el protocolo OBEX en Linux a través de la pila de protocolos BlueZ. ***ObexPush*** permite la carga y descarga de archivos hacia o desde cualquier dispositivo que implemente el protocolo OBEX Object Push.

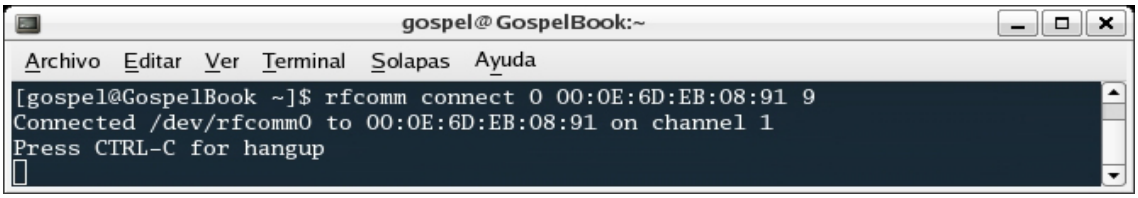


El procedimiento de transferencia de archivos con ***ObexPush*** transcurre de la siguiente forma. En primer lugar, se localiza el dispositivo Bluetooth al cual se quiere transferir el archivo con ayuda del comando ***hcitool scan***, luego se comprueba que soporta el Perfil de Carga de Objetos con ayuda del comando ***sdptool search OPUSH*** y finalmente se envía el archivo con la herramienta ***ObexPush***. Nótese que, en este caso, es necesario especificar el parámetro 9 que corresponde al canal que emplea OBEX Object Push en ese dispositivo.

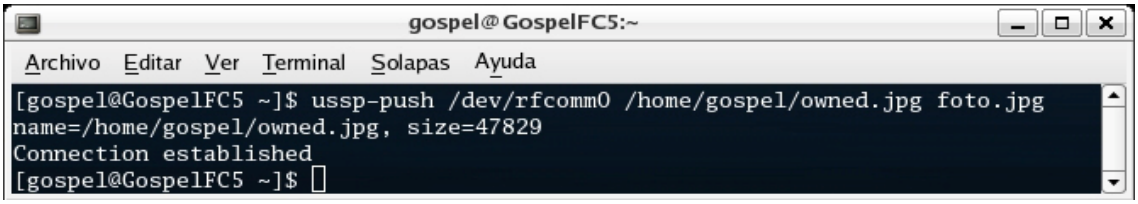
2.4.2.6.2 – Ussp-Push

Ussp-Push es otra herramienta basada en la pila de protocolos BlueZ para Linux. Permite únicamente la carga de objetos en otros dispositivos Bluetooth.

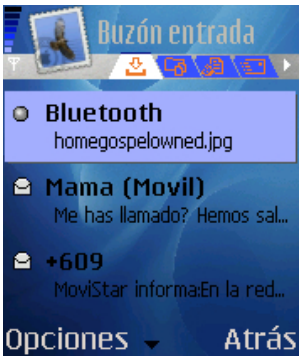
En las siguientes capturas, se muestra ***Ussp-Push*** en acción. En primer lugar, requiere establecer una conexión RFCOMM explícita con el dispositivo Bluetooth al cual se quiere transferir el archivo.



A continuación, desde otra ventana de shell lanzamos ***Ussp-Push***.



En los tres casos de ejemplo práctico de transferencia de archivos a través de aplicaciones basadas en el protocolo OBEX, el dispositivo servidor se trataba de un teléfono móvil Nokia™ 6600 basado en Symbian OS. Al recibir el archivo a través del protocolo OBEX, el teléfono lo almacena en la bandeja de entrada con formato de mensaje SMS y notifica al usuario de la recepción del mismo.



2.4.2.7 – Perfil de Sincronización

El Perfil de Sincronización define los requisitos para los protocolos y procedimientos utilizados por las aplicaciones que proporcionan el modelo de uso de *sincronización*. El modelo proporciona sincronización dispositivo a dispositivo de programas de gestión de la información personal (PIM, Personal Information Management). La información que manejan estos programas consiste normalmente en una agenda de teléfonos de contactos, calendario, mensajes y notas.

Los dispositivos que implementan el Perfil de Sincronización pueden actuar como cliente y servidor.

Las unidades activas en el modelo de uso de *sincronización* deben soportar tres funciones: sincronización, comando de sincronización y sincronización automática.

- La sincronización en Bluetooth debe soportar al menos una de las siguientes clases de aplicación:

    - Sincronización de agendas telefónicas
    - Sincronización de calendarios
    - Sincronización de mensajes
    - Sincronización de notas

Para conseguir la interoperabilidad a nivel de aplicación, se definen formatos de contenido específicos para cada unidad activa. Estos formatos de contenido son los siguientes: *vCard*, *vCalendar*, *vMessage* y *vNote*.

- La función de comando de sincronización permite a un dispositivo cliente trabajar como un servidor y recibir un comando de sincronización desde otro dispositivo cliente.

- La función conocida como sincronización automática permite a un dispositivo cliente iniciar la sincronización cuando el dispositivo servidor entra dentro de su rango de cobertura. En el nivel de banda base, esto significa que el cliente realiza una búsqueda del servidor a intervalos regulares, y cuando detecta que éste ha entrado en su rango de cobertura comienza la sincronización.

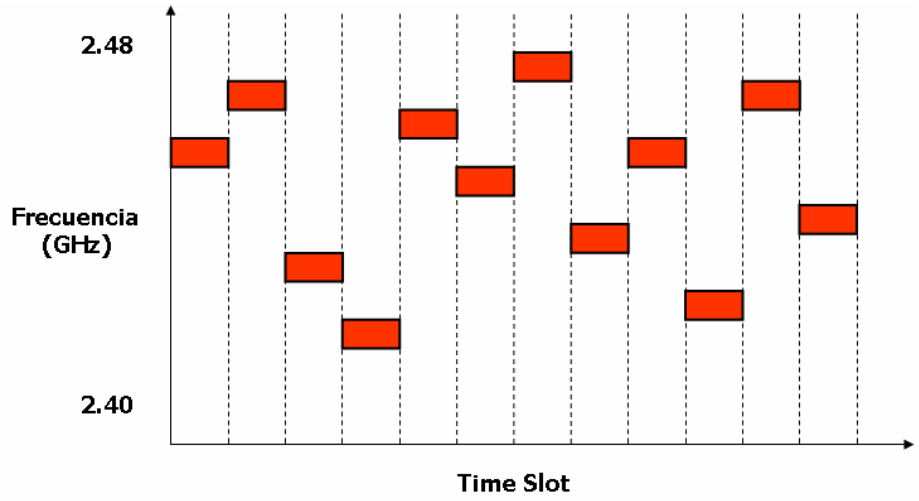
2.5 – Elementos de seguridad en Bluetooth

Bluetooth incorpora varios mecanismos de seguridad que lo convierten en uno de los protocolos de comunicaciones más seguros y robustos frente a ataques y capturas de datos. Se definen mecanismos de seguridad en las siguientes capas de protocolo:

- Seguridad a nivel de banda base
- Seguridad a nivel de enlace

2.5.1 – Seguridad a nivel de banda base

Bluetooth trabaja en la frecuencia de 2.4 GHz de la banda ISM (Industrial, Scientific and Medical) disponible a nivel mundial y que no requiere licencia de operador. Con el fin de evitar interferencias con otras tecnologías que operen en la misma banda de frecuencias, Bluetooth emplea la técnica de salto de frecuencias (FHSS, Frequency Hopping Spread Spectrum), que consiste en dividir la banda en 79 canales (23 en España, Francia y Japón) de longitud 1 MHz y realizar 1600 saltos por segundo.



Durante el proceso de establecimiento de la conexión en una piconet, el dispositivo maestro genera una tabla pseudoaleatoria con la secuencia o patrón de saltos de frecuencia que deben utilizar los dispositivos pertenecientes a la piconet durante las comunicaciones. El intercambio de la tabla de saltos desde el maestro hacia el esclavo (o esclavos) se realiza en un canal determinado del espectro de frecuencias, de forma que todos los dispositivos pueden acceder a ésta.

Cuando se establece la piconet, el dispositivo esclavo recibe un paquete FHS (Frecuency Hop Synchronization) que le permite sincronizar su reloj interno con el reloj del maestro agregando un desplazamiento a su reloj interno. Como los relojes funcionan con independencia, a lo largo de la comunicación se han de actualizar regularmente dichos desplazamientos.

Una vez comenzada la comunicación, el intercambio de paquetes de datos se realiza de acuerdo con el patrón de saltos de frecuencia establecido y a una velocidad marcada por el reloj interno. Esto significa que en cada instante de tiempo cada dispositivo escribirá o escuchará durante su timeslot en un determinado canal del espectro.

Cualquier dispositivo ajeno que no pertenezca a la piconet no podrá participar en la comunicación enviando paquetes o escuchando tráfico, ya que no dispone de la tabla con la secuencia de saltos utilizada en la piconet y, además, la probabilidad de adivinar cual de todos los canales puede ser empleado para la comunicación en cada instante de tiempo es mínima.

En definitiva, la técnica de saltos de frecuencia empleada por Bluetooth garantiza, en principio, la participación exclusiva de dispositivos autorizados en una piconet y una comunicación libre de escuchas por parte de usuarios ajenos a la misma.

Sin embargo, si un atacante pudiera disponer de la tabla de frecuencias generada por el dispositivo maestro de una piconet, éste podría sincronizar su módulo Bluetooth con el resto de dispositivos de la piconet y participar en la comunicación, capturando el tráfico e inyectando paquetes. Puesto que el intercambio de las tablas de secuencias de saltos se lleva a cabo en una frecuencia conocida, un dispositivo malicioso podría estar escuchando constantemente y capturar estas tablas de saltos de frecuencia para sincronizarse con una piconet.

Por otro lado, aunque actualmente los módulos Bluetooth convencionales no permiten la escucha en más de un canal del espectro de frecuencias, es posible que exista determinado hardware especializado capaz de barrer todo el espectro de frecuencias empleado por Bluetooth y capturar paquetes en más de un canal a la vez. Esto permitiría a un atacante escuchar todo el intercambio de paquetes entre dispositivos de una piconet y comprometer la confidencialidad de la comunicación. No obstante, se trata de módulos Bluetooth especiales que no resultan accesibles para la mayoría de los usuarios y cuyo coste es elevado.

Se puede concluir, por tanto, que la técnica de saltos de frecuencia empleada por Bluetooth refuerza en gran medida la seguridad del protocolo, pero en ningún caso garantiza totalmente la privacidad de la comunicación, ya que cabe la posibilidad de que cierto usuario no autorizado pueda conseguir acceso a la piconet y comprometa la confidencialidad del intercambio de datos y la integridad de los dispositivos participantes en la misma. Sin embargo, se trata de técnicas de ataque muy complejas y difíciles de llevar a la práctica en un entorno real.

2.5.2 – Seguridad a nivel de enlace

Se definen tres mecanismos de seguridad en el nivel de enlace

- Autenticación
- Autorización
- Cifrado de datos

2.5.2.1 – Autenticación

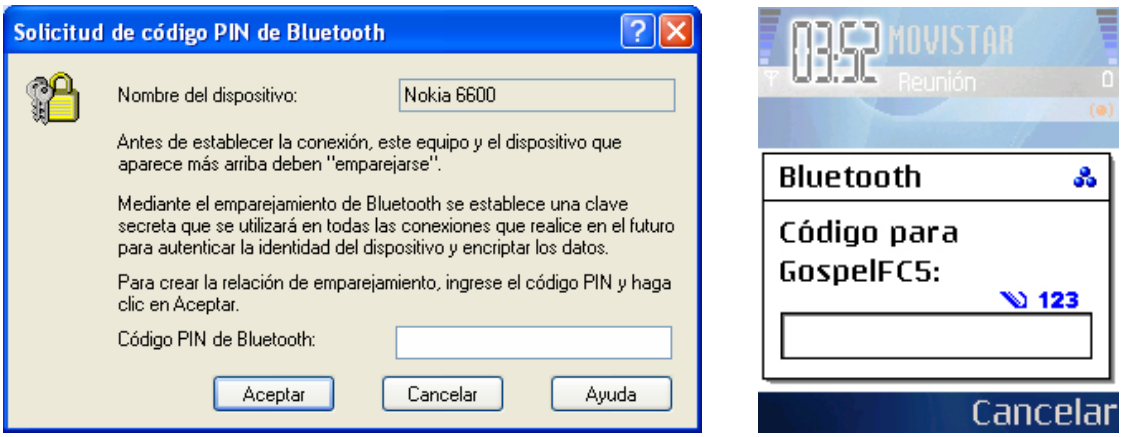
La autenticación es el proceso por el cual un dispositivo Bluetooth verifica su identidad en otro dispositivo para poder acceder a los servicios que ofrece.

Todas las funciones de seguridad de nivel de enlace están basadas en el concepto de claves de enlace, las cuales son números pseudoaleatorios de 128 bits almacenados individualmente por cada par de dispositivos Bluetooth. La autenticación no requiere la intervención del usuario; implica un esquema de desafío/respuesta entre cada par de dispositivos que emplea una clave de enlace secreta común de 128 bits. Consecuentemente, este esquema se utiliza para autentificar dispositivos, no usuarios.

La primera vez que dos dispositivos intentan comunicarse, se utiliza un procedimiento de inicialización denominado *emparejamiento* (pairing) para crear una clave de enlace común de una forma segura. Para la primera conexión entre dos dispositivos, el procedimiento estándar de emparejamiento requiere que el usuario de cada dispositivo introduzca un código (cadena ASCII) de seguridad Bluetooth de hasta 16 bytes de longitud que debe ser el mismo en los dos casos. En primer lugar un usuario introduce el código de seguridad y en segundo lugar, el otro usuario debe confirmar el mismo código de seguridad.

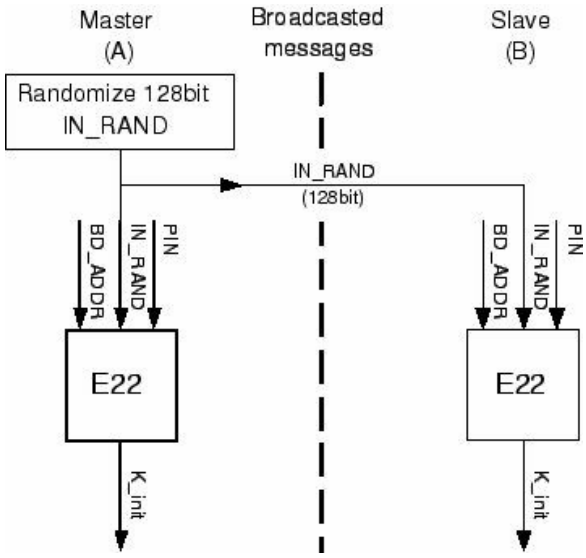
En el caso de requisitos de baja seguridad, es posible que aquellos dispositivos Bluetooth que no tengan interfaz de usuario para permitir al usuario introducir un código manualmente, como por ejemplo dispositivos GPS o manos libres, incorporen un código prefijado de fábrica por defecto, como 0000, 1234, etc. En este caso, en el dispositivo que inicia el emparejamiento se debe introducir el mismo código prefijado que incluye el dispositivo de baja seguridad.

El código de seguridad Bluetooth, a menudo es conocido como clave PIN (Personal Identification Number), aunque no se trata de un código que el usuario deba memorizar para mantenerlo en secreto, ya que se introduce una sola vez.



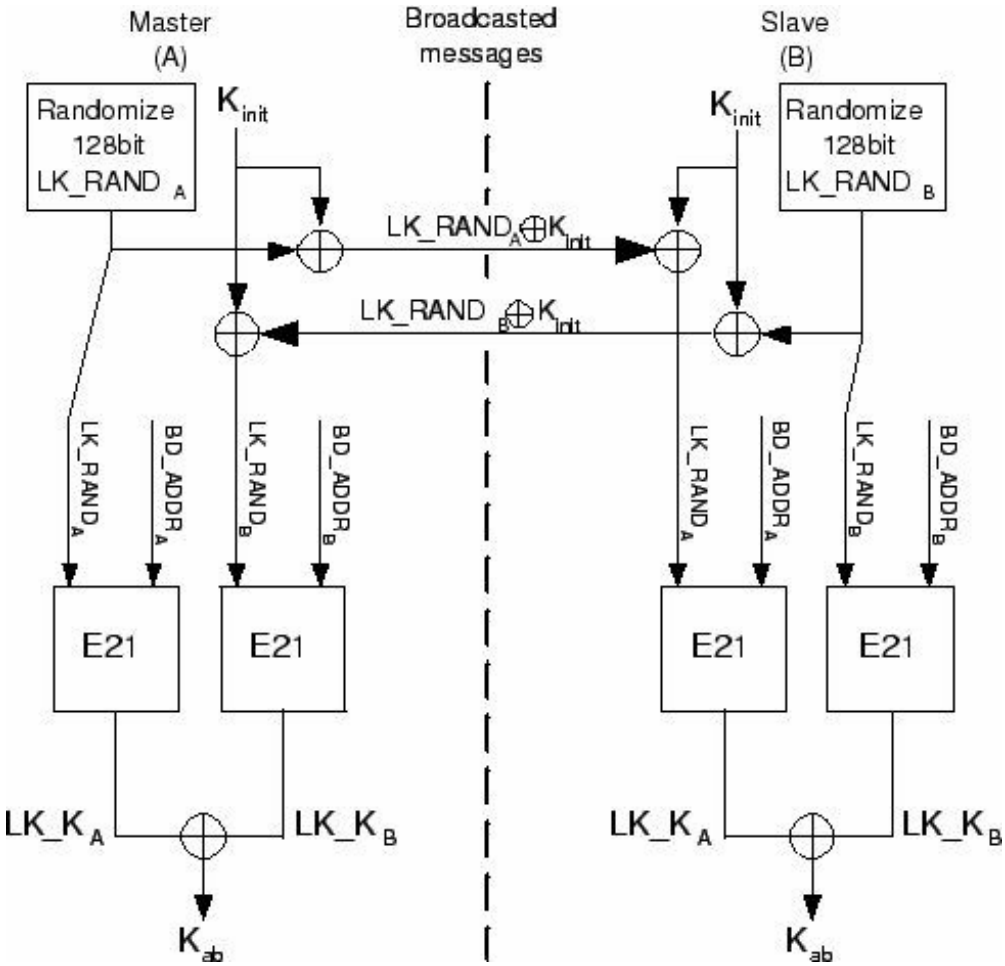
A partir del código de seguridad Bluetooth (PIN), se obtiene la clave de enlace común a dos dispositivos del siguiente modo:

1) Se genera una clave de inicialización común K_{init} de 128 bits usando el algoritmo E22 a partir del código de seguridad Bluetooth (PIN), la longitud del mismo, la dirección BD_ADDR de 48 bits y un número aleatorio *IN_RAND*.



2) Se genera la clave de enlace K_{ab} usando el algoritmo E21. Los dispositivos usan la clave de inicialización K_{init} para intercambiar dos nuevos números aleatorios de 128 bits, conocidos como LK_RAND A y LK_RAND B. Cada dispositivo genera un número aleatorio y se lo envía al otro dispositivo previamente XOReado bit a bit con K_{init}. Dado que ambos dispositivos conocen K_{init}, cada dispositivo conoce ambas LK_RAND.

   A partir de la dirección BD_ADDR y el LK_RAND, el algoritmo E21 obtiene la clave de enlace K_{ab}.



3) La clave de enlace común se almacena temporalmente en los dispositivos emparejados. Mientras esta clave de enlace esté almacenada en ambos dispositivos, no es necesario repetir el emparejamiento en futuras conexiones. Si por alguna razón, uno de los dos dispositivos ha borrado la clave de enlace común, debe repetirse el emparejamiento y los usuarios deben introducir de nuevo cualquier código de seguridad Bluetooth.

Una vez que los dispositivos emparejados disponen de la clave de enlace, utilizan esta clave común para autentificarse automáticamente en las sucesivas conexiones. El proceso de autenticación está basado en el esquema desafío/respuesta y transcurre de la siguiente forma:

1) El dispositivo reclamante envía su dirección BD_ADDR al dispositivo verificador.

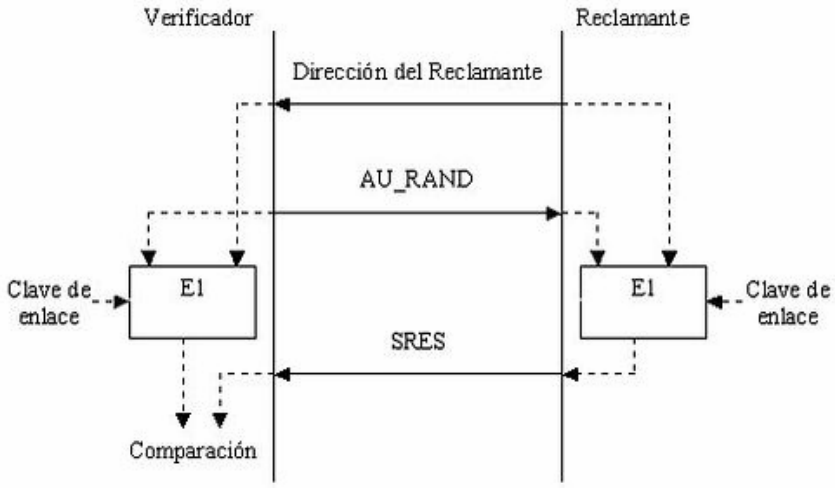
2) El verificador devuelve un desafío aleatorio de 128 bits al demandante.

3) El reclamante usa el algoritmo E1 para generar la respuesta de autenticación (SRES) de 32 bits, usando como parámetros de entrada la dirección BD_ADDR del reclamante, la clave de enlace K_{ab} almacenada y el desafío. El verificador realiza la misma operación en paralelo.

4) El reclamante devuelve la respuesta SRES al verificador.

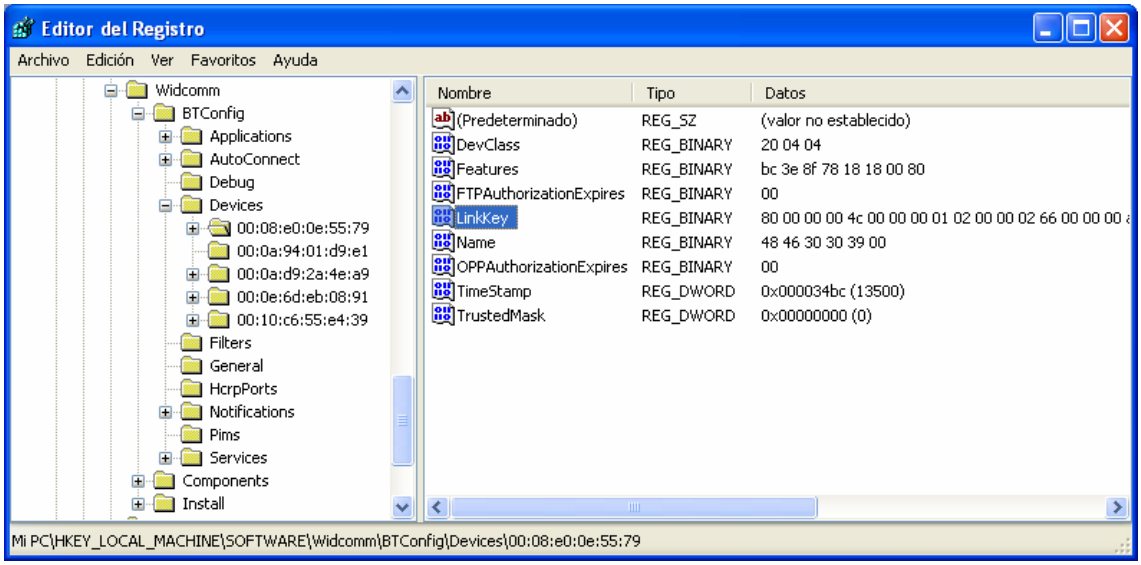
5) El verificador comprueba la respuesta SRES recibida por el reclamante con la respuesta SRES calculada por él.

6) Si los valores de SRES coinciden, el verificador establece la conexión.

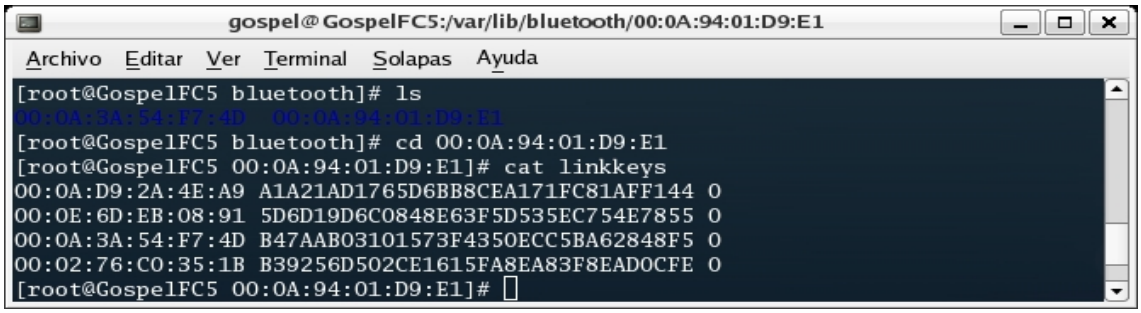


La especificación de Bluetooth establece que si se produce un fallo durante el proceso de autenticación, y para prevenir que un atacante pruebe claves de enlace aleatorias en un ataque de fuerza bruta, debe transcurrir cierto período de espera antes de que se pueda llevar a cabo un nuevo intento de autenticación. Para cada sucesivo intento fallido, el tiempo de espera aumenta exponencialmente.

En Microsoft Windows™, las claves de enlace generadas durante el emparejamiento con otros dispositivos Bluetooth se almacenan en el registro.



Linux almacena las claves de enlace en el archivo /var/lib/bluetooth/BD_ADDR/linkkeys, siendo BD_ADDR la dirección MAC del módulo Bluetooth local.



2.5.2.2 – Autorización

La autorización es el procedimiento que determina los derechos que tiene un dispositivo Bluetooth para acceder a los servicios que ofrece un sistema.

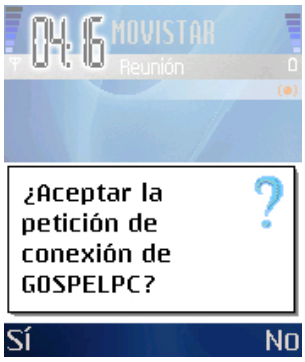
El mecanismo de autorización en dispositivos Bluetooth se lleva a cabo mediante *niveles de confianza*. Los dispositivos tienen tres niveles de confianza, los cuales determinan la capacidad de acceso a los servicios: total, parcial o restringida y nula.

- Un dispositivo de confianza mantiene una relación de emparejamiento y dispone de acceso sin restricciones a todos los servicios.

- Un dispositivo de confianza restringida mantiene una relación de emparejamiento y sólo dispone de acceso restringido a uno o varios servicios, pero no a todos.

- Un dispositivo no confiable es aquel que puede o no mantener tener una relación de emparejamiento pero que no es de confianza. No se le permite el acceso a ningún servicio.

En el caso de que un determinado dispositivo de confianza intente acceder a un servicio autorizado, no se requiere ningún procedimiento de confirmación, accede de forma transparente.

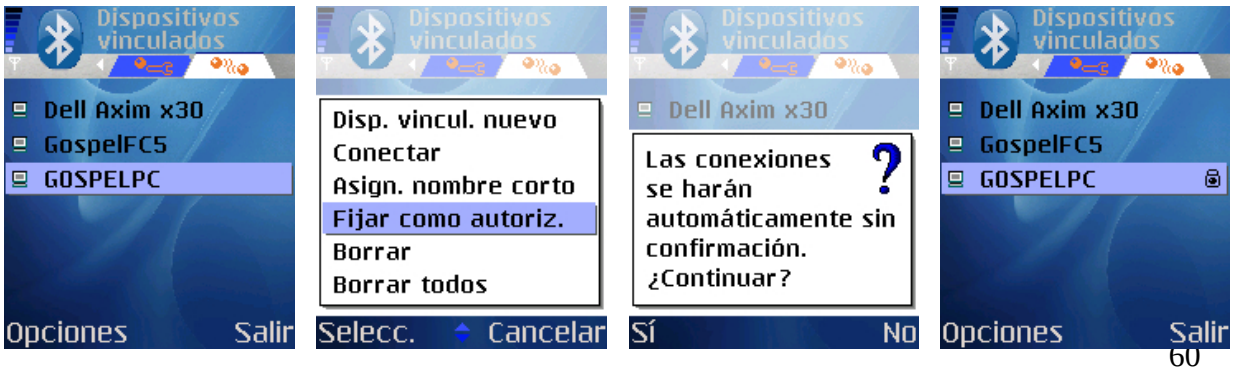


En el caso de que un determinado dispositivo no confiable intente acceder a un servicio restringido, se requiere un procedimiento explícito de confirmación por parte del usuario para permitir o denegar el acceso a ese dispositivo durante la sesión de conexión actual. Nótese que, para algunos servicios, es posible conceder permisos de acceso temporal a dispositivos no emparejados previamente.

Todo dispositivo Bluetooth dispone de una base de datos interna con su lista de dispositivos de confianza. Esa base de datos tiene el siguiente formato:

Campo	**Estado**	**Contenido**
BD_ADDR	Obligatorio	Dirección MAC del dispositivo
Nivel de confianza	Obligatorio	De confianza / No de confianza
Clave de enlace	Obligatorio	Clave de enlace K_{ab}
Nombre	Opcional	Nombre del dispositivo (Cadena)

En la mayoría de dispositivos es posible configurar manualmente la lista de dispositivos de confianza, por ejemplo, en un Nokia™ 6600.



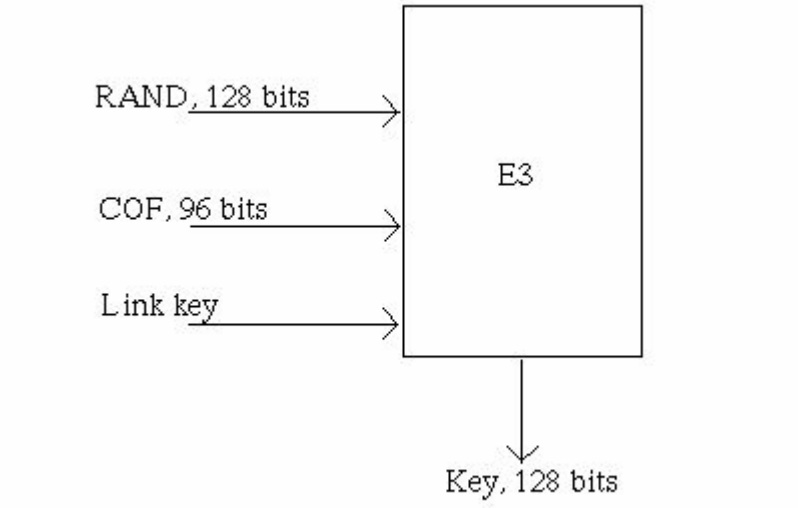
2.5.2.3 – Cifrado de datos

El cifrado de datos protege la información que se transmite en un enlace entre dispositivos Bluetooth. Garantiza la confidencialidad del mensaje transmitido, de forma que si el paquete es capturado por un usuario que no posea la clave de descifrado, el mensaje le resultará ininteligible.

Su implementación es opcional, pero necesita que se haya producido anteriormente una autenticación. El maestro y el esclavo deben ponerse de acuerdo en utilizar cifrado o no. En caso afirmativo, deben determinar el tamaño de la clave de cifrado, para lo cual, maestro y esclavo intercambian mensajes hasta alcanzar un acuerdo. No siempre es posible llegar a un acuerdo sobre el tamaño de la clave, en este caso se indica a las unidades Bluetooth que no se les permite comunicarse utilizando cifrado en el enlace.

Tras esta negociación comienza el proceso de cifrado:

El maestro genera una clave de cifrado K_C de 128 bits usando el algoritmo E3, el cual requiere como parámetros de entrada un número aleatorio de 128 bits, la clave de enlace K_{AB} generada durante el procedimiento de emparejamiento y número COF (Ciphering Offset) de 96 bits basado en el valor temporal ACO (Authenticated Ciphering Offset) calculado durante el procedimiento de autenticación.



Una vez que la clave de cifrado se ha generado con éxito, el maestro se encuentra en condiciones de transmitir datos cifrados, para lo cual debe detener temporalmente el tráfico de datos de los niveles superiores y así evitar la recepción de datos corruptos.

2.5.2.4 – SAFER+

SAFER+ (Secure And Fast Encryption Routine) es un algoritmo simétrico de cifrado de datos de tipo IBC (Iterated Block Ciphers) que utiliza bloques de 128 bits.

Bluetooth hace uso del algoritmo de cifrado SAFER+ durante la generación de las claves de autenticación y de cifrado:

- Generación de la clave K_{init} con el algoritmo E22
- Generación de la clave de enlace K_{ab} con el algoritmo E21
- Proceso de autenticación con el algoritmo E1
- Generación de la clave de cifrado K_c con el algoritmo E3

Sin embargo, SAFER+ no es utilizado para cifrar el enlace de datos. Para esta función, Bluetooth utiliza el algoritmo 4LFSR, que es un cifrador de flujo adecuado para cifrado rápido de datos.

2.5.2.5 – Modos de seguridad Bluetooth a nivel de enlace

Una vez descritos los 3 mecanismos que emplea Bluetooth para reforzar la seguridad a nivel de enlace (autenticación, autorización y cifrado de datos), se definen 3 modos de seguridad a nivel de enlace en función de la implementación de los mismos:

- Modo 1: Ausencia de seguridad. Todos los mecanismos de seguridad (autenticación y cifrado) están deshabilitados. Además, el dispositivo se sitúa en modo *promiscuo*, permitiendo que todos los dispositivos Bluetooth se puedan conectar a él. Este modo lo emplean dispositivos que no tienen aplicaciones críticas.

  Ninguna parte del tráfico de datos es cifrada.

- Modo 2: Proporciona seguridad en los servicios a nivel de L2CAP. Utiliza mecanismos de seguridad después de establecerse el canal de comunicación: autorización. Un gestor de seguridad controla el acceso de los dispositivos a los diferentes servicios, en función de su nivel de confianza.

  La interacción con el usuario se limita a solicitar confirmación de la autorización de acceso a servicios restringidos por parte de dispositivos no autorizados.

  El tráfico de difusión no está cifrado, mientras que el tráfico punto a punto se cifra según las claves individuales generadas durante la conexión.

- Modo 3: Proporciona seguridad en el dispositivo a nivel de LMP. Utiliza mecanismos de seguridad antes de establecerse el canal de comunicación: autenticación. Requiere emparejamiento de dispositivos y existencia de clave de enlace compartida para validar la conexión entre dispositivos.

  La interacción con el usuario requiere la introducción de un código PIN para llevar a cabo el emparejamiento de dispositivos.

  Todo el tráfico se cifra con la clave de cifrado generada.

Se ha visto como se definen dos tipos de mecanismos de control de acceso a los servicios soportados por un dispositivo: autenticación y autorización. Esto significa que existen algunos servicios únicamente accesibles mediante autenticación y otros a los que se podría acceder mediante simple autorización, sin necesidad de que los dispositivos hayan sido emparejados previamente.

La implementación de los distintos modos de seguridad a nivel de enlace se ha ido incorporando progresivamente en la fabricación de teléfonos móviles Bluetooth de la siguiente manera:

Los primeros modelos de teléfonos móviles Bluetooth que aparecieron en el mercado fueron Nokia™ 6310 y Sony-Ericsson™ T68 y T610. Las primeras versiones de estos dispositivos incorporaban por defecto el Modo 1 de seguridad de enlace, lo que permitía a cualquier usuario de otro dispositivo acceder a todos los servicios y establecer conexiones con los perfiles Bluetooth soportados sin necesidad de autenticación ni autorización. Evidentemente, esto trajo consigo la aparición de los primeros ataques a teléfonos móviles: conexiones RFCOMM sin necesidad de autenticación con la posibilidad de ejecutar comandos AT en los terminales comprometidos.

Inmediatamente, los fabricantes se dieron cuenta del enorme riesgo que suponía no proteger los teléfonos móviles. Las sucesivas versiones que aparecieron en el mercado comenzaron a incorporar el Modo de seguridad de enlace 2 y, más tarde, el Modo 3. Esta implementación más robusta de los modos de seguridad en los servicios soportados por los teléfonos móviles fue realizándose de forma paulatina, primero en los servicios más críticos y, por último, en todos los servicios. Esto ocurrió así porque se descubrió que era posible acceder a servicios protegidos utilizando como puente conexiones a servicios no protegidos, por lo que resultaba más seguro proteger inicialmente todos los servicios.

Todavía, hoy en día, es posible encontrar teléfonos móviles vulnerables a ataques a servicios que no requieren autenticación y en los cuales es factible saltar la barrera de la autorización con ayuda de ingeniería social, engañando al usuario para que permita al dispositivo atacante acceder a cierto servicio.

Sin embargo, en líneas generales, casi la totalidad de los teléfonos móviles existentes en el mercado actual incorporan el modo 3 de seguridad a nivel de enlace en todos los servicios soportados, a excepción del Perfil de Carga de Objetos (OBEX Object Push), el cual sólo requiere autorización para permitir así el libre intercambio de tarjetas de visita entre dispositivos Bluetooth.

Capítulo

3

IDENTIFICACIÓN DE DISPOSITIVOS BLUETOOTH

3.1 – La pila de protocolos BlueZ para Linux

BlueZ es la pila de protocolos Bluetooth oficial para Linux. Inicialmente desarrollado por Qualcomm, ahora es un proyecto *open source* distribuido bajo licencia GPL (GNU General Public License). El núcleo de BlueZ forma parte del kernel oficial de Linux desde la versión 2.4.6.

La página oficial del proyecto BlueZ es http://www.bluez.org

El núcleo de BlueZ viene acompañado por un conjunto de herramientas que permiten ejecutar las funciones Bluetooth implementadas en la pila de protocolos desde una shell de comandos.
Estas herramientas son las siguientes:

- ***Bluepin***: Gestión de suministro del PIN para emparejamiento con otros dispositivos.

- ***Hciattach***: Configuración de dispositivos serial UART (Universal Asynchronous Receiver/Transmitter) como interfaces HCI Bluetooth.

- ***Hciconfig***: Configuración de dispositivos Bluetooth locales.

- ***Hcid***: Demonio interfaz HCI

- ***Hcidump***: *Sniffer* local de tráfico HCI que entra y sale por el dispositivo Bluetooth instalado en el sistema.

- ***Hcitool***: Gestión del enlace con otros dispositivos Bluetooth, detección de dispositivos remotos, resolución de nombres, identificación de clases, etc.

- ***L2ping***: Envío de solicitudes *echo request* (pings) a nivel L2CAP.

- ***Pand***: Gestión de conexiones PAN (Personal Area Network)

- ***Rfcomm***: Gestión de conexiones RFCOMM

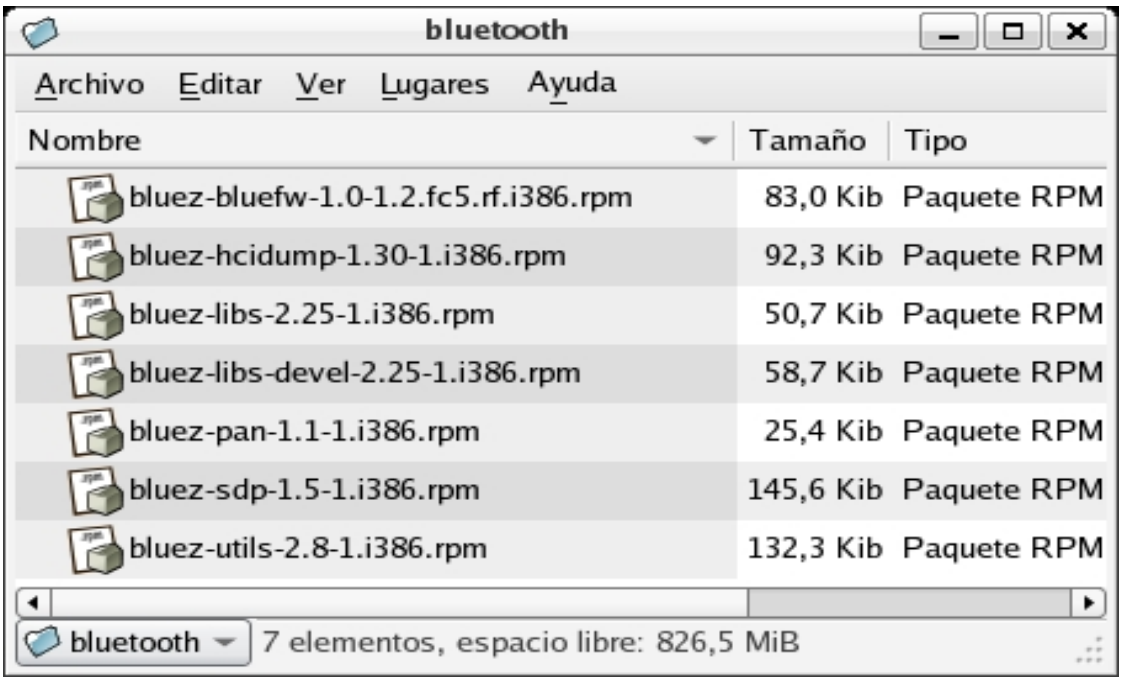
- ***Sdpd***: Demonio del protocolo de descubrimiento de servicios SDP. Se encarga de proporcionar acceso a los servicios Bluetooth locales.

- ***Sdptool***: Gestión de SDP (Service Discovery Protocol), descubrimiento de servicios Bluetooth en dispositivos remotos.

La mayoría de las herramientas mencionadas se encuentran instaladas por defecto en aquellas distribuciones Linux que incorporan el núcleo de BlueZ. Sin embargo, también es posible obtener las herramientas y librerías necesarias para el funcionamiento de BlueZ por medio de módulos del núcleo BlueZ. Estos módulos se encuentran disponibles para descarga en la página web oficial de Bluez http://www.bluez.org/download.html y son los siguientes:

- bluez-libs-x.x.tar.gz (Librerías básicas Bluetooth)
- bluez-libs-devel-x.x.tar.gz (Librerías de desarrollo Bluetooth)
- bluez-utils-x.x.tar.gz (Herramientas Bluetooth)
- bluez-firmware-x.x.tar.gz (Actualización de *firmware*)
- bluez-hcidump-x.x.tar.gz (*Sniffer* local de tráfico HCI)

Así mismo, también es posible obtener estos módulos en forma de paquetes precompilados dependientes de cada distribución.



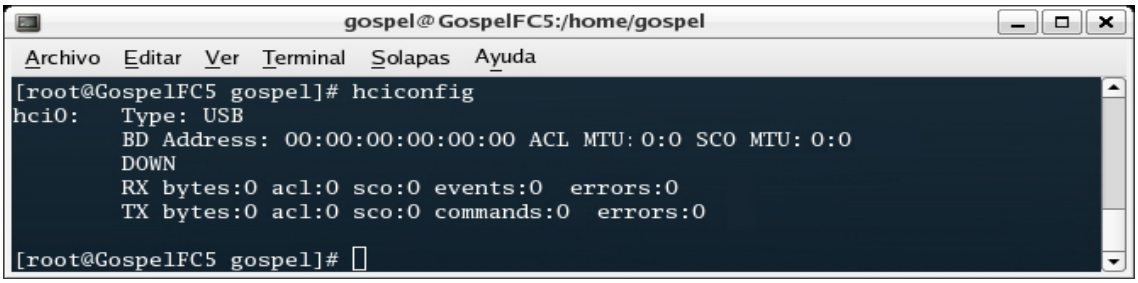
3.2 – Interconexión con dispositivos Bluetooth desde Linux

La pila de protocolos Bluetooth para Linux, BlueZ, permite conectar un PC con dispositivos Bluetooth remotos. Las diferentes herramientas que incluye permiten detectar dispositivos Bluetooth cercanos, obtener información básica de los mismos y conectarse a los servicios que soportan.

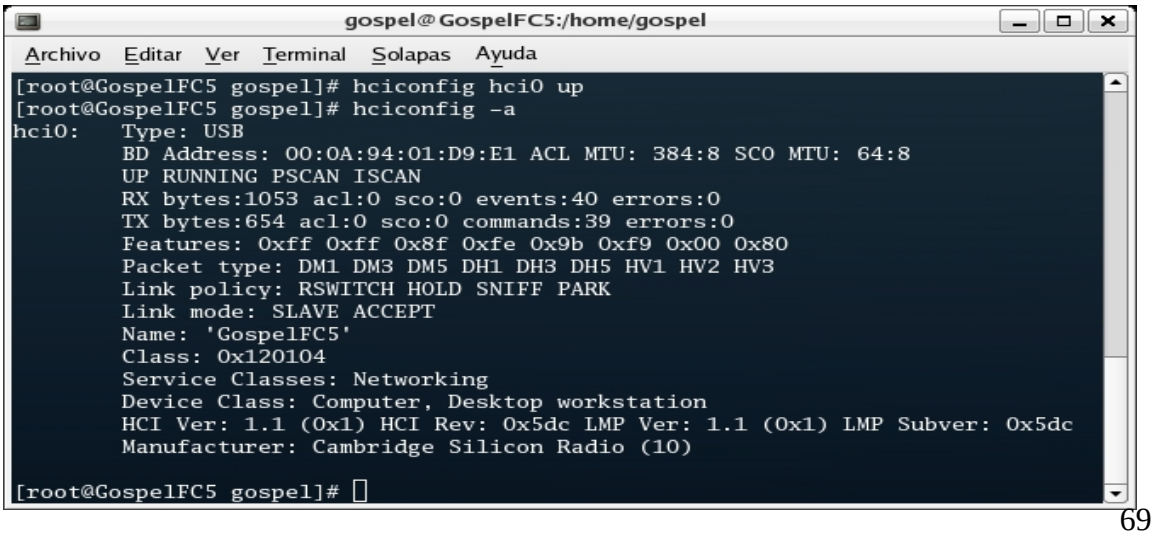
3.2.1 – Configuración del dispositivo Bluetooth local

El primer paso es conectar al PC el módulo Bluetooth que se va a emplear para la comunicación con otros dispositivos. Linux debería reconocer automáticamente el dispositivo sin necesidad de instalar *drivers*. No obstante, es posible que algún módulo Bluetooth requiera la instalación adicional de *drivers* en el sistema antes de utilizarlo. En tal caso, se debe consultar con el fabricante.

Así mismo, lo más habitual es que Linux monte automáticamente en el interfaz hci0 el módulo Bluetooth conectado, pero en algunas distribuciones puede no suceder. La verificación se realiza mediante la herramienta ***Hciconfig***.



En caso de que el dispositivo Bluetooth no se haya montado automáticamente, será necesario montarlo manualmente con la herramienta ***Hciconfig***.



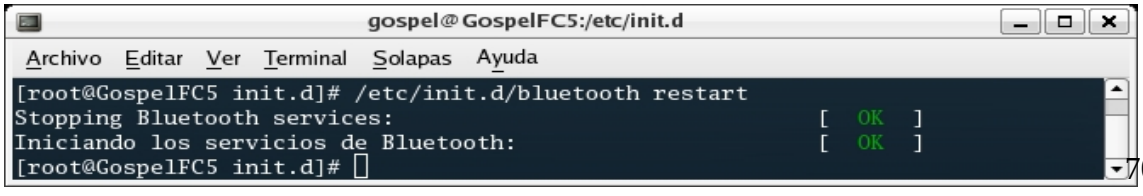
3.2.2 – Configuración de opciones del interfaz HCI

Antes de establecer comunicación con otro dispositivo Bluetooth, se debe configurar el fichero de opciones de HCI, localizado en /etc/bluetooth/hcid.conf.

Es recomendable utilizar la siguiente configuración para hcid.conf:

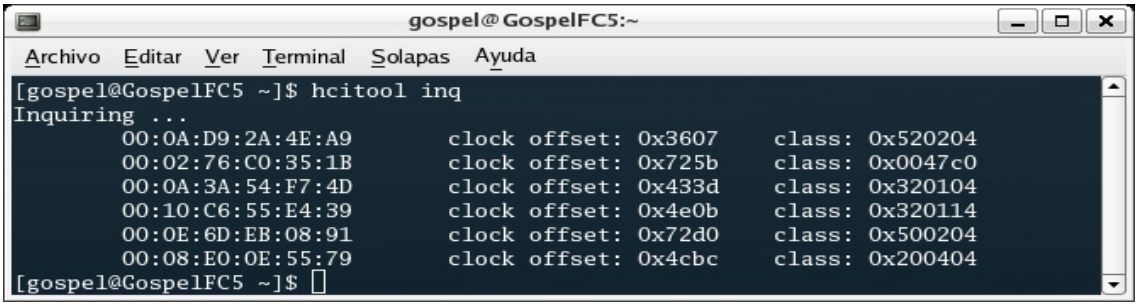
```
# HCId options
options {
        # Automatically initialize new devices
        autoinit yes;

        # Security Manager mode
        #   none - Security manager disabled
        #   auto - Use local PIN for incoming connections
        #   user - Always ask user for a PIN
        #
        security auto;

        # Pairing mode
        #   none  - Pairing disabled
        #   multi - Allow pairing with already paired devices
        #   once  - Pair once and deny successive attempts
        pairing multi;

        # PIN helper
        pin_helper /usr/bin/bluepin;
}

# Default settings for HCI devices
device {
        # Local device name
        #   %d - device id
        #   %h - host name
        name "GospelFC5";

        # Local device class
        class 0x120104;

        # Inquiry and Page scan
        iscan enable; pscan enable;

        # Default link mode
        #   none   - no specific policy
        #   accept - always accept incoming connections
        #   master - become master on incoming connections,
        #            deny role switch on outgoing connections
        lm accept;

        # Default link policy
        #   none    - no specific policy
        #   rswitch - allow role switch
        #   hold    - allow hold mode
        #   sniff   - allow sniff mode
        #   park    - allow park mode
        lp rswitch,hold,sniff,park;

        # Authentication and Encryption (Security Mode 3)
        #auth enable;
        #encrypt enable;
}
```

Tras aplicar los cambios, procede a la reinicializacion de los servicios Bluetooth.

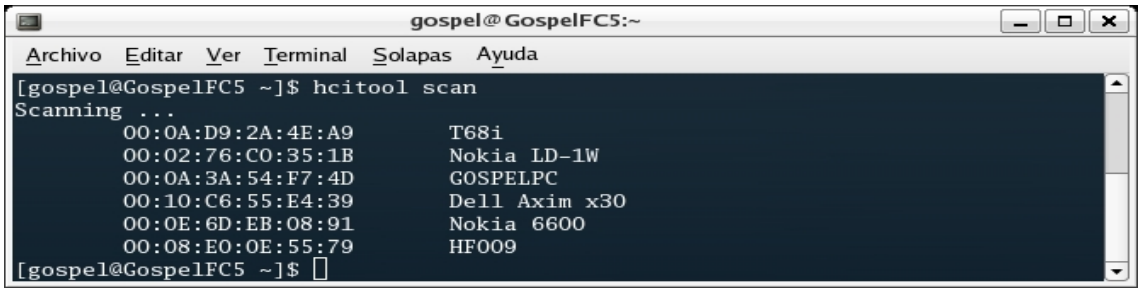


3.2.3 – Detección de dispositivos Bluetooth con *Hcitool*

La herramienta ***Hcitool*** permite enviar paquetes *inquiry* para detectar la existencia de dispositivos Bluetooth cercanos.

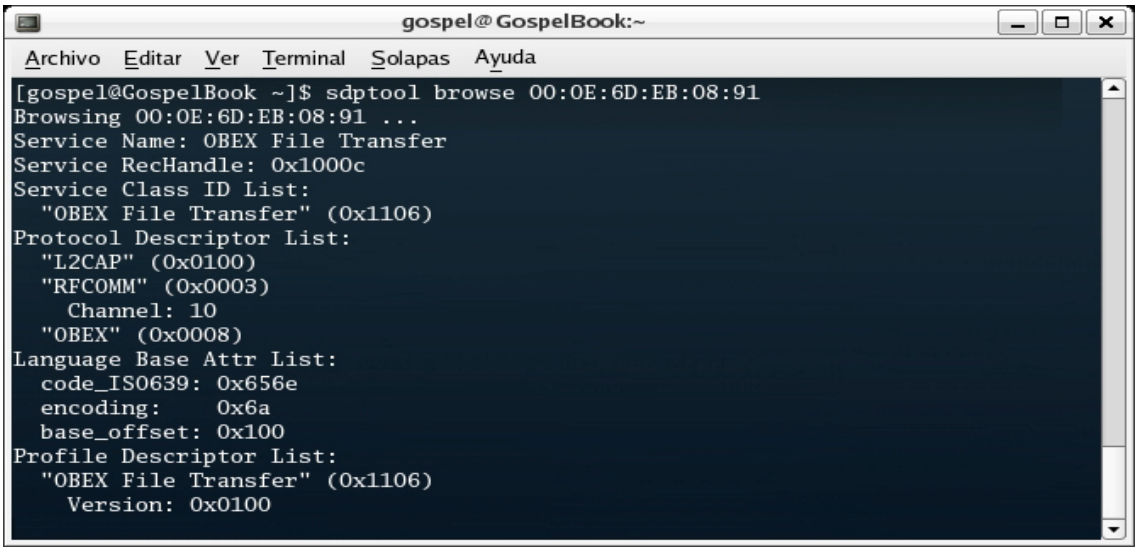


Así mismo, también es posible obtener cierta información sobre los dispositivos detectados, como su *Class of Device/Service* y su nombre de dispositivo.



3.2.4 – Descubrimiento de servicios Bluetooth con *Sdptool*

La herramienta ***Sdptool*** permite identificar los perfiles disponibles en un dispositivo Bluetooth detectado, por ejemplo, un teléfono móvil.

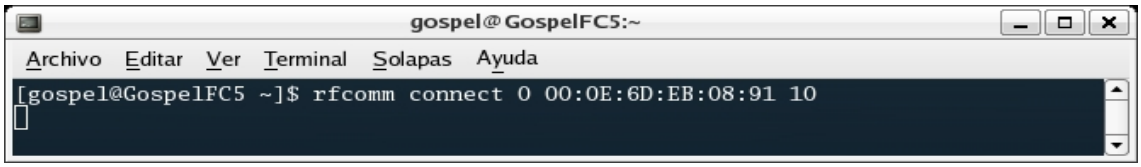



gospel@GospelFC5:~
Archivo Editar Ver Terminal Solapas Ayuda
Service Name: Dial-up Networking
Service RecHandle: 0x10001
Service Class ID List:
"Dialup Networking" (0x1103)
"Generic Networking" (0x1201)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel: 1
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
Profile Descriptor List:
"Dialup Networking" (0x1103)
Version: 0x0100
Service Name: Bluetooth Serial Port
Service Description: Bluetooth Serial Port
Service Provider: Symbian Ltd.
Service RecHandle: 0x10002
Service Class ID List:
"Serial Port" (0x1101)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel: 2
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
Service Name: OBEX Object Push
Service RecHandle: 0x10003
Service Class ID List:
"OBEX Object Push" (0x1105)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel: 9
"OBEX" (0x0008)
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
Profile Descriptor List:
"OBEX Object Push" (0x1105)
Version: 0x0100
Service Name: Handsfree Audio Gateway
Service RecHandle: 0x1000d
Service Class ID List:
"Handfree Audio Gateway" (0x111f)
"Generic Audio" (0x1203)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel: 3
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
Profile Descriptor List:
"Handsfree" (0x111e)
Version: 0x0101
[gospel@GospelFC5 ~]$

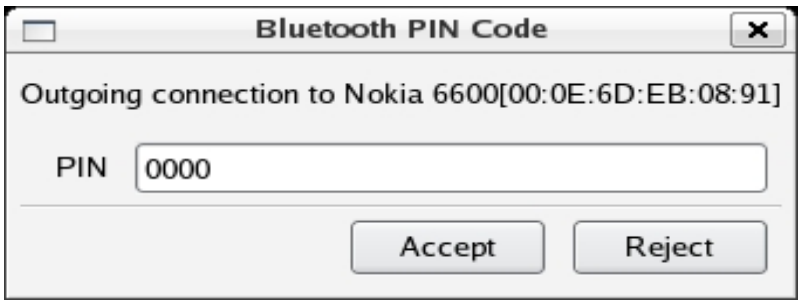
3.2.5 – Conexión a un servicio de otro dispositivo

Sdptool permite obtener una lista de perfiles soportados por un dispositivo Bluetooth remoto, que incluye información detalla sobre las características de cada perfil, incluyendo el canal asociado.

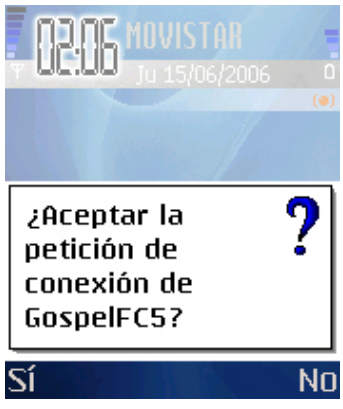
Con ayuda de la herramienta ***Rfcomm***, es posible conectarse a un determinado servicio conociendo la dirección BD_ADDR y el canal destino. Como ejemplo, se describe la conexión al Perfil de Transferencia de Archivos (OBEX File Transfer) del teléfono móvil, cuyo canal correspondiente es el 10.



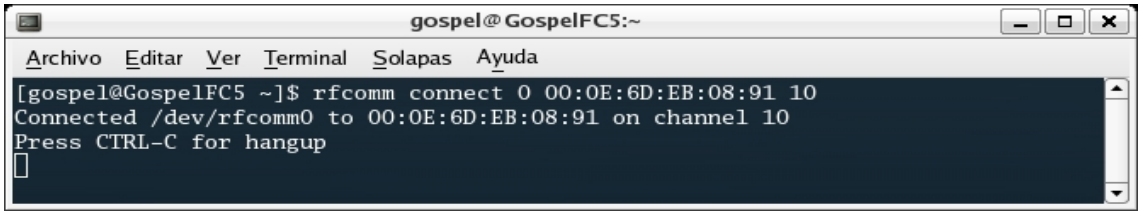
El intento de conexión inicia el procedimiento de **autenticación**. En el teléfono móvil aparecerá una notificación de emparejamiento de dispositivos Bluetooth, que requerirá la introducción de un código PIN. Tras la confirmación, aparecerá una ventana en el interfaz de Linux solicitando la introducción del mismo código PIN al usuario.



Si el código PIN coincide, los dispositivos quedan emparejados y se establece la conexión a nivel de enlace. Sin embargo, antes de poder acceder al servicio, hay que superar con éxito el procedimiento de **autorización**. En este caso, el dispositivo Bluetooth conectado al PC con Linux no estaba incluido en la lista de dispositivos de confianza del teléfono móvil, por lo que se hace necesario que el usuario del teléfono móvil confirme la autorización de acceso del dispositivo remoto al servicio OBEX File Transfer.



Tras la confirmación de la autorización, Linux puede acceder al servicio y establecer la conexión a nivel de aplicación.



3.3 – BlueZScanner: el escáner de dispositivos Bluetooth

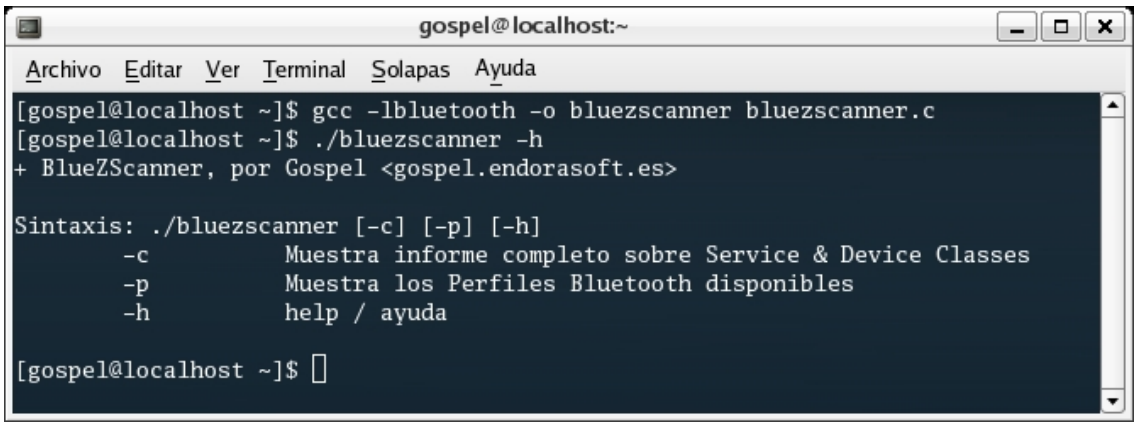
BlueZScanner es un sencillo escáner de dispositivos Bluetooth que utiliza BlueZ y está desarrollado en lenguaje C. ***BlueZScanner*** forma parte de este Proyecto Fin de Carrera "Seguridad en Bluetooth".

Implementa las siguientes funciones:

- Detección de dispositivos Bluetooth cercanos.
- Resolución de nombre de dispositivo.
- Obtención del fabricante del chip Bluetooth incorporado en el dispositivo.
- Análisis del campo Device Class, que identifica la naturaleza del dispositivo.
- Análisis de los campos Service Classes, que identifican los servicios ofrecidos por el dispositivo.
- Descubrimiento de Perfiles Bluetooth disponibles en el dispositivo.

El código fuente y la herramienta ***BlueZScanner*** están disponibles para descarga en la siguiente dirección: http://gospel.endorasoft.es/

El código fuente de ***BlueZScanner*** se distribuye libremente bajo licencia GNU. Se necesita instalar previamente el paquete de librerías *bluez-libs-devel*.



En los sucesivos capítulos se explicarán conceptos avanzados sobre la programación Bluetooth en C con entorno BlueZ y se mostrarán ejemplos prácticos llevados a cabo con la herramienta ***BlueZScanner***.

3.4 – Detección de dispositivos Bluetooth

3.4.1 – Detección de dispositivos en modo visible o *discoverable*

El proceso de detección de dispositivos Bluetooth forma parte de las funciones de la capa HCI.

- Inicialmente, el dispositivo origen envía paquetes *inquiry* y se mantiene en espera de recibir respuestas de otros dispositivos presentes en su zona de cobertura.

- Si los dispositivos destino están configurados en modo visible (*discoverable*) se encontrarán en estado *inquiry_scan* y en predisposición de atender estos paquetes. En este caso, al recibir un paquete *inquiry* cambiarán a estado *inquiry_response* y enviarán una respuesta al dispositivo origen con sus direcciones MAC y otros parámetros.

- Los dispositivos que estén configurados en modo no visible (*non discoverable*) se encontrarán en modo *inquiry_response* y, por tanto, no responderán al dispositivo origen y permanecerán ocultos.

El siguiente código fuente para BlueZ permite escanear y detectar dispositivos Bluetooth en modo visible:

```
#include <stdio.h>
#include <stdlib.h>
#include <bluetooth/bluetooth.h>
#include <bluetooth/hci.h>
#include <bluetooth/hci_lib.h>

int main ()
{
   inquiry_info *ii = NULL; //Almacena la lista de dispositivos
                       //detectados durante el inquiry

   int max_rsp, num_rsp; //Nº de respuestas/dispositivos detectados
   int dev_id; //Identificador del adaptador Bluetooth local
   int socket; //Socket HCI;
   int len, i;

   char MAC_dev[20]; //Direccion MAC del dispositivo detectado
   char nombre_dev[248]; //Nombre del dispositivo detectado
```

```
   //Obtenemos el identificador del adaptador local Bluetooth
   dev_id = hci_get_route(NULL);
   if (dev_id < 0)
   {
      printf("Error. Dispositivo Bluetooth local no disponible.\n");
      exit(1);
   }

   //Abrimos un socket local HCI
   socket = hci_open_dev(dev_id);
   if (socket < 0)
   {
      printf("Error. Fallo al intentar abrir socket HCI.\n");
      exit(1);
   }

   //Inicializamos algunas variables
   len = 8; //El tiempo de inquiry es de 1.28x8=10.24 secs/dispositivo

   max_rsp = 255; //Se pueden detectar a lo sumo 255 dispositivos

   //Creamos la lista de dispositivos detectados con hci_inquiry
   ii = (inquiry_info*)malloc(max_rsp * sizeof(inquiry_info));

   printf("Detectando dispositivos...\n\n");

   //hci_inquiry lleva a cabo un descubrimiento de dispositivos
   //Bluetooth y devuelve una lista de dispositivos detectados en
   //inquiry_info ii para ser almacenados.
   //La bandera IREQ_CACHE_FLUSH permite que la caché sea limpiada
   //antes de buscar nuevos dispositivos, ya que podrian aparecer
   //dispositivos anteriormente detectados pero ahora fuera de rango.

   num_rsp    =    hci_inquiry(dev_id,    len,    max_rsp,    NULL,    &ii,
IREQ_CACHE_FLUSH);

   if(num_rsp < 0)
      printf("Error. Fallo al intentar hci_inquiry.\n");

   //Para cada una de las respuestas obtenidas durante el inquiry
   //obtenemos el nombre del dispositivo

   for(i=0;i<num_rsp;i++)
   {
         ba2str(&(ii+i)->bdaddr, MAC_dev);
         memset(nombre_dev, 0, sizeof(nombre_dev));
         if(hci_read_remote_name(socket,                &(ii+i)->bdaddr,
sizeof(nombre_dev), nombre_dev, 0) < 0)
         {
            strcpy(nombre_dev, "[Desconocido]");
         }

      printf("Dispositivo (%d) encontrado:\n\tMAC: %s\t\tNombre: %s\n\n",
i+1, MAC_dev, nombre_dev);
   }

   free(ii);
   close(socket);
   return(0);
}
```

BlueZScanner se basa en parte del anterior código fuente.

La siguiente captura muestra como ***BlueZScanner*** detecta dispositivos Bluetooth cercanos:

3.4.2 – Detección de dispositivos en modo oculto o *non discoverable*

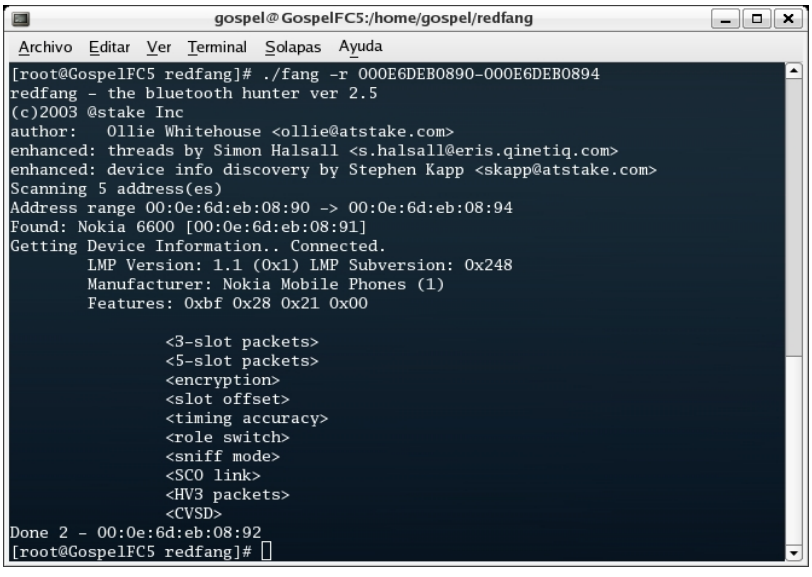
Como ya se ha comentado anteriormente, durante el proceso de HCI *inquiry*, los dispositivos configurados en modo oculto no responderán a los paquetes *inquiry* del dispositivo origen.

Sin embargo, el hecho de no ver un dispositivo oculto, no significa que no esté ahí y que no se pueda detectar con otros medios.

Existen ciertos paquetes Bluetooth que, enviados a una dirección MAC determinada, obligan al dispositivo que los recibe a devolver una respuesta independientemente del modo de visibilidad establecido. Una de estos paquetes es el utilizado en la función de resolución de nombre de dispositivo *hci_read_remote_name*. Esta función, implementada en BlueZ, permite obtener el nombre de un dispositivo Bluetooth a partir de su dirección MAC. Es decir, conociendo la dirección MAC de un dispositivo Bluetooth remoto podemos resolver su nombre con la función *hci_read_remote_name*, aunque el dispositivo haya sido configurado en modo oculto e ignore paquetes HCI *inquiry*.

La técnica que permite descubrir dispositivos Bluetooth en modo oculto, dado que, a priori, la dirección MAC del dispositivo es desconocida, consiste en realizar un ataque por fuerza bruta sobre un determinado rango de direcciones MAC permitidas esperando encontrar alguna que responda a la petición de resolución de nombre de dispositivo, lo que indicaría que existe un dispositivo Bluetooth configurado en modo oculto detrás de esa dirección MAC.

Esta misma técnica es la empleada por ***RedFang***, una herramienta desarrollada por *@stake Inc*. que permite encontrar dispositivos Bluetooth en modo oculto realizando fuerza bruta sobre los 6 bytes de la dirección BD_ADDR Bluetooth.



3.5 – Descubrimiento de perfiles Bluetooth

Como se ha explicado anteriormente, el descubrimiento de perfiles en dispositivos Bluetooth remotos corre a cargo del protocolo SDP (Service Discovery Protocol).

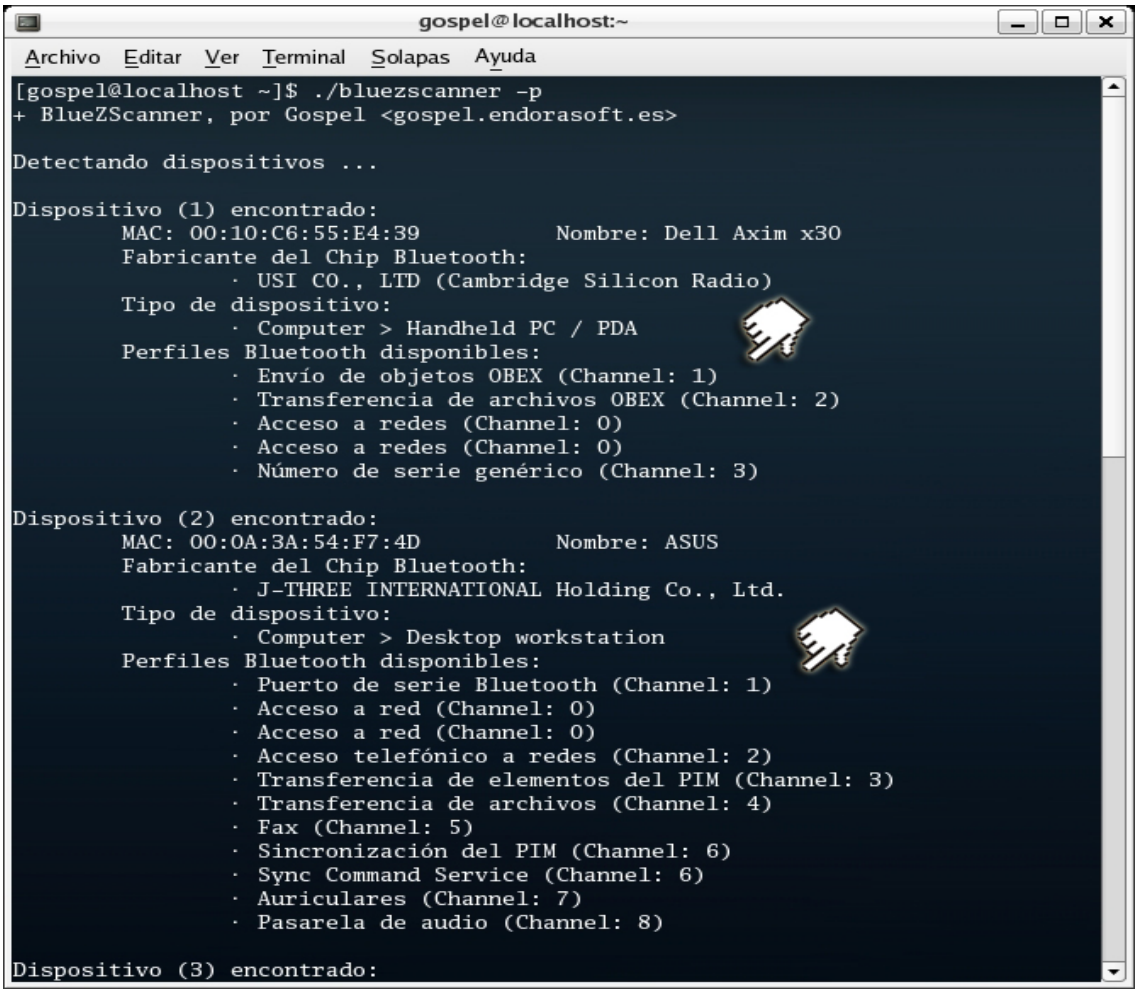
El siguiente código fuente para BlueZ permite obtener los perfiles Bluetooth soportados por un dispositivo con una determinada dirección MAC:

```
void getServicios(char MAC_dev[])
{
   bdaddr_t bdaddr;
   sdp_list_t *attrid, *search, *seq;
   uint32_t range = 0x0000ffff;
   sdp_session_t *sess;
   struct hci_dev_info di;
   uuid_t root_uuid;

   if(hci_devinfo(0, &di) < 0)
   {
      printf("[!] Error. Fallo en HCI device info.\n");
      exit(1);
   }

   str2ba(MAC_dev,&bdaddr);

   sess = sdp_connect(&di.bdaddr, &bdaddr, SDP_RETRY_IF_BUSY);

   if(!sess)
   {
      printf("Error. Imposible conectar con el servidor SDP.\n");
      exit(1);
   }

   printf("\tPerfiles Bluetooth disponibles:\n", MAC_dev);

   sdp_uuid16_create(&root_uuid, PUBLIC_BROWSE_GROUP);
   attrid = sdp_list_append(0, &range);
   search = sdp_list_append(0, &root_uuid);

   if(sdp_service_search_attr_req(sess,    search,    SDP_ATTR_REQ_RANGE,
attrid, &seq) < 0)
   {
      perror("SDP service search");
      sdp_close(sess);
      exit(1);
   }

   sdp_list_free(attrid, 0);
   sdp_list_free(search, 0);

   //Imprimimos la lista de Perfiles Bluetooth encontrados
   for(; seq; seq = seq->next)
   {
      sdp_record_t *rec = (sdp_record_t *) seq->data;
      sdp_list_t *access = NULL;
```

```
        int channel;
        sdp_data_t *d = sdp_data_get(rec, SDP_ATTR_SVCNAME_PRIMARY);
        if (d)
        {
           printf("\t\t· %s ", d->val.str);
        }

        sdp_get_access_protos(rec, &access);

        //Obtenemos el channel correspondiente al Perfil
        if(access)
        {
           channel = sdp_get_proto_port(access, RFCOMM_UUID);
           printf("(Channel: %d)\n", channel);
        }
     }
     free(seq);
     sdp_close(sess);
}
```

BlueZScanner se basa en parte del anterior código fuente.

La siguiente captura muestra como ***BlueZScanner*** es capaz de descubrir perfiles Bluetooth en dispositivos detectados:



3.6 – Identificación del tipo de dispositivo Bluetooth

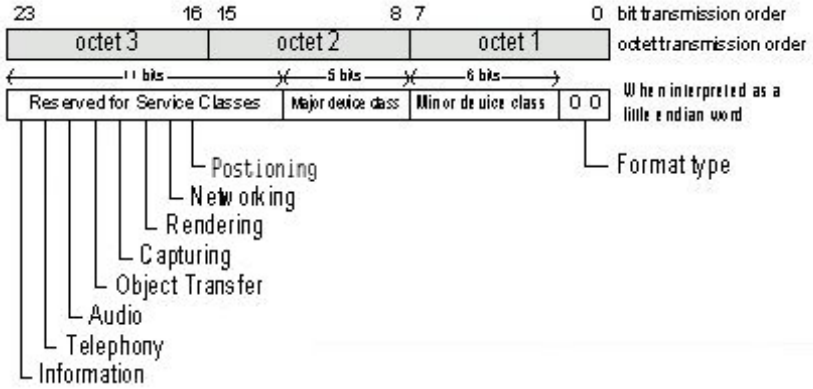
Cuando se realiza un escaneo de dispositivos Bluetooth con ayuda de herramientas comerciales o, simplemente, con el asistente de conexiones Bluetooth de Microsoft Windows™, los dispositivos detectados se muestran mediante iconos representativos de su naturaleza, ya sean PCs, PDAs, Teléfonos móviles, Manos libres, etc.

El reconocimiento se efectúa a través de los **DIACs (The General- and Device-Specific Inquiry Access Codes)**

Cada dispositivo Bluetooth incorpora en la cabecera de nivel de Banda Base de sus paquetes un campo **Class of Device/Service**. Este campo se compone de 3 octetos organizados con el siguiente formato:

- 11 últimos bits reservados para las **Service Classes**.
- 11 bits siguientes reservados para las **Device Classes**.
  - 6 bits reservados para las **Major Device Classes**.
  - 5 bits reservados para las **Minor Device Classes**.
- 2 primeros bits para el campo Format Type, por defecto a 0.

El siguiente esquema resume lo explicado:



Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

El campo reservado para las *Device Classes* permite identificar la naturaleza del dispositivo. Este campo se compone 2 subcampos: *Major Device Class* y *Minor Device Class*.

3.6.1 – *Major Device Class*

El campo reservado para la *Major Device Class* permite identificar el tipo genérico de dispositivo. Este campo se compone de 5 bits, desde el bit 8 al 12. Cada tipo genérico de dispositivo está asociado a una representación concreta de bits dentro del campo. En la especificación de banda base 1.1 de Bluetooth se describe la siguiente correspondencia entre bits marcados en el campo *Major Device Class* y los tipos genéricos de dispositivos:

12	11	10	9	8	Major Device Class
0	0	0	0	0	Miscellaneous [Ref #2]
0	0	0	0	1	Computer (desktop,notebook, PDA, organizers, ...)
0	0	0	1	0	Phone (cellular, cordless, payphone, modem, ...)
0	0	0	1	1	LAN /Network Access point
0	0	1	0	0	Audio/Video (headset,speaker,stereo, video display, vcr.....
0	0	1	0	1	Peripheral (mouse, joystick, keyboards, ...)
0	0	1	1	0	Imaging (printing, scanner, camera, display, ...)
0	0	1	1	1	Wearable
0	1	0	0	0	Toy
1	1	1	1	1	Uncategorized, specific device code not specified
X	X	X	X	X	All other values reserved

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

3.6.2 – *Minor Device Class*

El campo reservado para la Minor Device Class permite identificar el tipo específico de dispositivo. Este campo se compone de 6 bits, desde el bit 7 al 2. Cada tipo específico de dispositivo está asociado a una representación concreta de bits dentro del campo. En la especificación de banda base 1.1 de Bluetooth se describe la siguiente correspondencia entre bits marcados en el campo *Minor Device Class* y los tipos específicos de dispositivos, dentro de cada tipo genérico.

Computer Major Class

7	6	5	4	3	2	Minor Device Class bit no of CoD
0	0	0	0	0	0	Uncategorized, code for device not assigned
0	0	0	0	0	1	Desktop workstation
0	0	0	0	1	0	Server-class computer
0	0	0	0	1	1	Laptop
0	0	0	1	0	0	Handheld PC/PDA (clam shell)
0	0	0	1	0	1	Palm sized PC/PDA
0	0	0	1	1	0	Wearable computer (Watch sized)
X	X	X	X	X	X	All other values reserved

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

Phone Major Class

7	6	5	4	3	2	Minor Device Class bit no of CoD
0	0	0	0	0	0	Uncategorized, code for device not assigned
0	0	0	0	0	1	Cellular
0	0	0	0	1	0	Cordless
0	0	0	0	1	1	Smart phone
0	0	0	1	0	0	Wired modem or voice gateway
0	0	0	1	0	1	Common ISDN Access
X	X	X	X	X	X	All other values reserved

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

Peripheral Major Class

7	6	Minor Device Class bit no of CoD
0	0	Not Keyboard / Not Pointing Device
0	1	Keyboard
1	0	Pointing device
1	1	Combo keyboard/pointing device

5	4	3	2	Minor Device Class bit no of CoD
0	0	0	0	Uncategorized device
0	0	0	1	Joystick
0	0	1	0	Gamepad
0	0	1	1	Remote control
0	1	0	0	Sensing device
0	1	0	1	Digitizer tablet
0	1	1	0	Card Reader (e.g. SIM Card Reader)
X	X	X	X	All other values reserved

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

Audio / Video Major Class

7	6	5	4	3	2	Minor Device Class bit no of CoD
0	0	0	0	0	0	Uncategorized, code not assigned
0	0	0	0	0	1	Wearable Headset Device
0	0	0	0	1	0	Hands-free Device
0	0	0	0	1	1	(Reserved)
0	0	0	1	0	0	Microphone
0	0	0	1	0	1	Loudspeaker
0	0	0	1	1	0	Headphones
0	0	0	1	1	1	Portable Audio
0	0	1	0	0	0	Car audio
0	0	1	0	0	1	Set-top box
0	0	1	0	1	0	HiFi Audio Device
0	0	1	0	1	1	VCR
0	0	1	1	0	0	Video Camera
0	0	1	1	0	1	Camcorder
0	0	1	1	1	0	Video Monitor
0	0	1	1	1	1	Video Display and Loudspeaker
0	1	0	0	0	0	Video Conferencing
0	1	0	0	0	1	(Reserved)
0	1	0	0	1	0	Gaming/Toy
X	X	X	X	X	X	All other values reserved

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

Imaging Major Class

7	6	5	4	Minor Device Class bit no of CoD
X	X	X	1	Display
X	X	1	X	Camera
X	1	X	X	Scanner
1	X	X	X	Printer
X	X	X	X	All other values reserved

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

LAN/Network Access Point Major Class – Load Factor

7	6	5	Minor Device Class bit no of CoD
0	0	0	Fully available
0	0	1	1 - 17% utilized
0	1	0	17 - 33% utilized
0	1	1	33 - 50% utilized
1	0	0	50 - 67% utilized
1	0	1	67 - 83% utilized
1	1	0	83 - 99% utilized
1	1	1	No service available

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

Weareable Major Class

7	6	5	4	3	2	Minor Device Class bit no of CoD
0	0	0	0	0	1	Wrist Watch
0	0	0	0	1	0	Pager
0	0	0	0	1	1	Jacket
0	0	0	1	0	0	Helmet
0	0	0	1	0	1	Glasses
X	X	X	X	X	X	All other values reserved

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

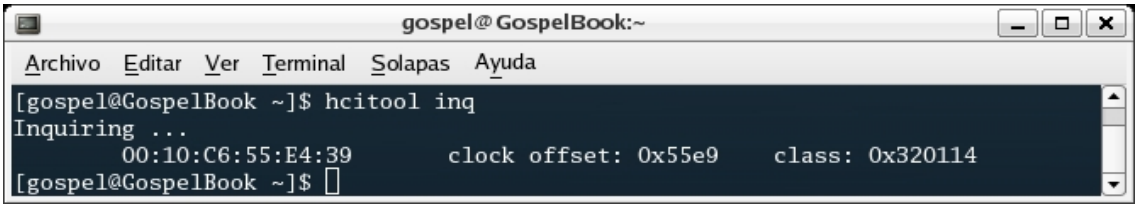
Toy Major Class

7	6	5	4	3	2	Minor Device Class bit no of CoD
0	0	0	0	0	1	Robot
0	0	0	0	1	0	Vehicle
0	0	0	0	1	1	Doll / Action Figure
0	0	0	1	0	0	Controller
0	0	0	1	0	1	Game
X	X	X	X	X	X	All other values reserved

Fuente: https://www.bluetooth.org/foundry/assignnumb/document/baseband

3.6.3 – Cálculo del *Class of Device*

Conociendo el *Class of Device/Service* de un dispositivo Bluetooth, se puede averiguar fácilmente el tipo de dispositivo del que se trata.



En este ejemplo, la herramienta ***Hcitool*** ha detectado un dispositivo cuyo *Class of Device/Service* es 0x320114.

La representación de 0x320114 en binario es la siguiente:

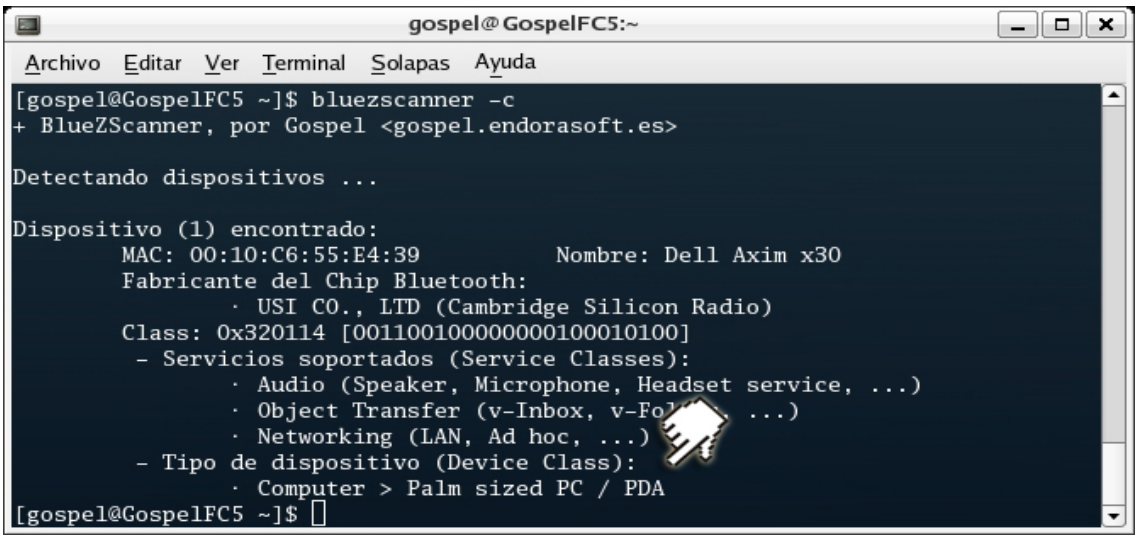
Nº bit:	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Valor:	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0

Se observa que el bit 8 aparece activado, que, tal y como establece la tabla *Major Device Classes*, corresponde al siguiente tipo de dispositivo genérico:

- bit 8: Computer (desktop,notebook, PDA, organizers, ...)

Al mismo tiempo, los bits 4 y 2 también aparecen activados que, tal y como establece la tabla *Minor Device Classes* de la categoría *Major Computer Class*, corresponden al siguiente tipo de dispositivo específico:

- bits 4 y 2: Palm sized PC / PDA



BlueZScanner implementa la función de identificación del tipo de dispositivos Bluetooth detectados, como se puede apreciar en la siguiente captura:

3.7 – Identificación del fabricante del chip Bluetooth

Todos los dispositivos Bluetooth se identifican mediante una dirección MAC, denominada BD_ADDR, que es unívoca y corresponde a un único módulo Bluetooth. Esta dirección es utilizada por el protocolo Bluetooth para el direccionamiento de dispositivos a nivel de red.

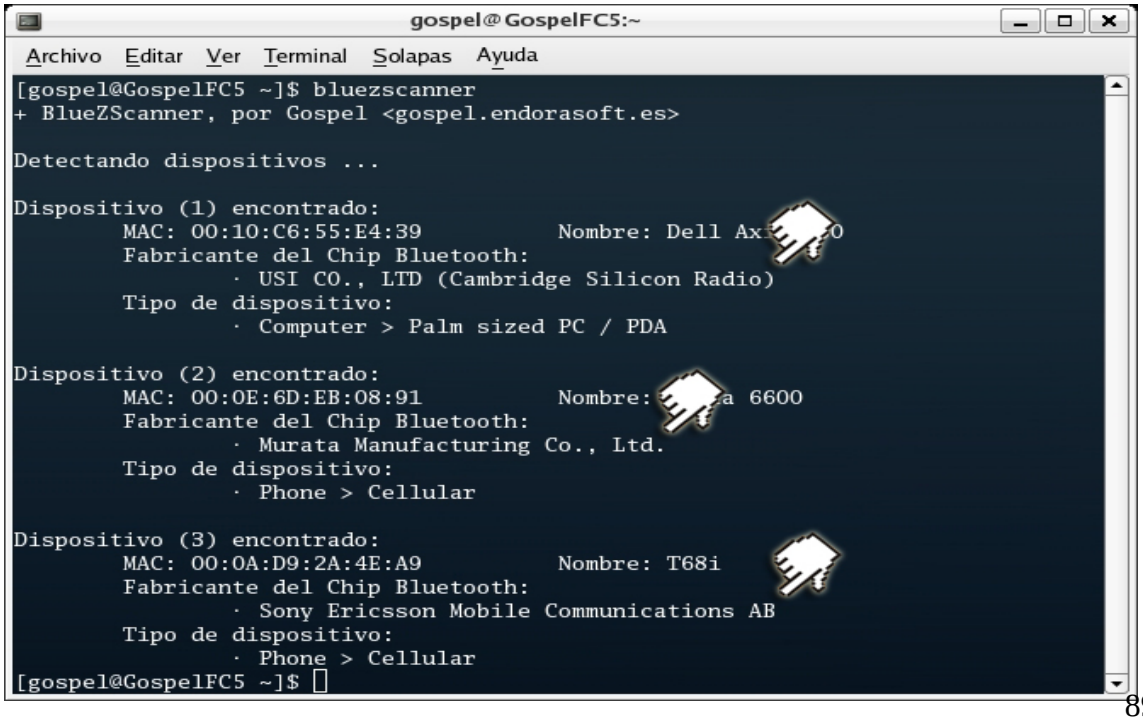
La dirección MAC de un dispositivo Bluetooth se compone de 6 bytes que responden a la siguiente notación: FF:FF:FF:XX:XX:XX. Los tres primeros bytes se asocian al fabricante del chip y los tres últimos identifican al dispositivo.

A continuación se presenta un ejemplo de relación entre los tres primeros bytes de la dirección MAC con el fabricante del chip:

```
00:0B:AC ..... 3Com Europe Ltd.
00:80:37 ..... Sony-Ericsson Group
00:0A:D9 ..... Sony Ericsson Mobile Communications AB
00:60:57 ..... Murata Manufacturing Co., Ltd. (Nokia)
```

Una lista más extensa de códigos MAC asignados a fabricantes de productos IEEE 802 está disponible en: http://standards.ieee.org/regauth/oui/oui.txt

BlueZScanner incluye un algoritmo capaz de obtener el código de tres bytes de una dirección MAC de un dispositivo y encontrar el fabricante del chip Bluetooth asociado a ese código. De esta forma, aunque el nombre del dispositivo no aporte información, su dirección MAC permite identificar la marca de manufactura.



3.8 – Identificación de la marca y modelo de un dispositivo

En principio, no es posible identificar inequívocamente la marca y modelo de un dispositivo Bluetooth detectado atendiendo a la información pública que ofrece. En algunos casos, es posible identificar la marca y modelo de un dispositivo realizando peticiones a ciertos servicios protegidos que ofrece el mismo y que requieren autenticación, como a través de comandos AT (en teléfonos móviles) o servicios de Sync ML DM Device Description (en dispositivos que soporten este protocolo).

Sin embargo, se puede afirmar que cada dispositivo incorpora una firma personal (*fingerprint*) en la información que muestra en su perfil público. La información disponible de forma pública en un dispositivo Bluetooth es la siguiente:

- Campo *Class of Device/Service*
- Dirección MAC
- Perfiles Bluetooth ofrecidos por el dispositivo

Si se realiza un análisis conjunto de los datos obtenidos, es posible obtener una relación casi unívoca por cada dispositivo como se detalla a continuación:

- A través del campo Class of Device/Service se puede identificar el tipo y naturaleza del dispositivo, es decir, averiguar si se trata de un PC, de un teléfono móvil o de una PDA.

- Los tres primeros bytes de la dirección MAC del dispositivo identifican el fabricante del chip Bluetooth. En algunos casos, el nombre del fabricante permite una identificación explícita, como por ejemplo el código del fabricante "Sony Ericsson Mobile Communications AB", que corresponde a teléfonos móviles de marca Sony-Ericsson™. En otros casos, la identificación es implítica, como por ejemplo el código del fabricante "Murata Manufacturing Co., Ltd.", que habitualmente responde a teléfonos móviles de marca Nokia™.

- Cada dispositivo Bluetooth ofrece un determinado conjunto de perfiles Bluetooth. En algunos casos, dos teléfonos móviles de una misma marca pueden soportar distintos perfiles o, a pesar de soportar los mismos perfiles, difieren en el contenido de algunos campos informativos del *Service Record*.

Se puede calcular un resumen (hash) de la información obtenida al realizar una petición SDP (Service Discovery Protocol) en un dispositivo y asociarlo con la información del fabricante del chip para conseguir un registro identificativo (firma o *fingerprint*) del dispositivo en cuestión. Por lo general, un registro calculado con este procedimiento puede asociarse unívocamente a un único dispositivo (marca y modelo), aunque puede que un único dispositivo disponga de varios registros asociados al mismo, debido fundamentalmente a que existen distintas versiones en el *firmware* del modelo y cada una puede soportar un conjunto diferente de perfiles Bluetooth.

Blueprinting Hash	Manufacturer	Model	Firmware
08:00:17@2949325	HP	iPAQ 5500	PocketPC (4.20.1081)
00:0C:55@983040	Microsoft	Windows XP	SP2
C6:F7:4A@655407	Motorola	A1000	unknown
00:0A:28@1769675	Motorola	V600	unknown
00:60:57@1704044	Nokia	3650	unknown
00:60:57@1704020	Nokia	3650	unknown
00:60:57@1704022	Nokia	3650	unknown
00:60:57@1704023	Nokia	3650	unknown
00:60:57@3605290	Nokia	6310i	unknown
00:60:57@3607710	Nokia	6310i	unknown
00:60:57@1704035	Nokia	6600	unknown
00:60:57@1704034	Nokia	6600	unknown
00:02:EE@4391166	Nokia	6820	unknown

Fuente: http://trifinite.org/Downloads/Blueprinting.pdf

Para conseguir una base de datos suficientemente completa, sería necesario escanear multitud de dispositivos Bluetooth y obtener registros identificativos por coincidencia de resúmenes calculados sobre el conjunto de perfiles Bluetooth. Mediante esta base de datos sería posible identificar con una alta probabilidad un nuevo dispositivo Bluetooth detectado en base a la experiencia recogida en anteriores escaneos.

Actualmente, dos proyectos de herramientas Bluetooth incorporan un mecanismo de aprendizaje para obtener registros identificativos de dispositivos Bluetooth y emplear esa experiencia recogida para poder identificar nuevos dispositivos detectados con una probabilidad bastante elevada. Estos proyectos son:

- ***Blueprinting,*** de *Trifinite Group*): http://trifinite.org/
- ***XBlue***, de Endorasoft: http://www.endorasoft.es/

Capítulo

4

ATAQUES A DISPOSITIVOS BLUETOOTH

4.1 – Ataques a teléfonos móviles

Los teléfonos móviles son los dispositivos que incorporan tecnología Bluetooth más comercializados en todo el mundo. Según Nokia™: "El mercado de móviles Bluetooth ha crecido hasta llegar a los 133 millones de unidades vendidas en 2005, y se espera que este número aumente hasta llegar a los 220 millones de unidades en 2006". Estas cifras suponen casi la mitad de los teléfonos móviles disponibles en el mercado.

Teniendo en cuenta el enorme éxito de los teléfonos móviles Bluetooth, los primeros ataques se centraron en la explotación de vulnerabilidades presentes en estos dispositivos Bluetooth. Cabe destacar que estas vulnerabilidades se debían a la pobre implementación de los mecanismos de seguridad por parte de los fabricantes de teléfonos móviles con Bluetooth. No se trataba por tanto, de vulnerabilidades inherentes al protocolo Bluetooth en sí, ya que, como se ha visto, Bluetooth es uno de los estándares de comunicaciones más seguros y robustos.

Uno de los objetivos principales que persiguen los ataques a teléfonos móviles Bluetooth es la ejecución de comandos AT en el terminal comprometido, lo que permitiría a un atacante obtener el control total del dispositivo. Otros ataques a teléfonos móviles Bluetooth permiten extraer información sensible, como la agenda de contactos o los mensajes SMS (Short Message Service) almacenados.

Los primeros teléfonos móviles en incorporar tecnología Bluetooth fueron los modelos Sony-Ericsson™ T68/T68i y T610, Nokia™ 6310/6310i, 7650 y 8910 y Motorola™ v600.

La implementación de seguridad en estos primeros modelos era muy pobre, ya que incorporaban por defecto el Modo 1 de seguridad a nivel de enlace, lo que permitía a cualquier usuario de otro dispositivo acceder a todos los servicios y establecer conexiones con los perfiles Bluetooth soportados sin necesidad de autenticación ni autorización.

Con menos de un año de diferencia desde la comercialización del primer teléfono Bluetooth, se descubrieron las primeras vulnerabilidades y se desarrollaron los primeros ataques. Los fabricantes se dieron cuenta del enorme riesgo que suponía no proteger los teléfonos móviles, así que las sucesivas versiones comercializadas incorporaron el Modo 2 de seguridad a nivel de enlace y, más tarde, el Modo 3. Esta implementación más robusta de los modos de seguridad en los servicios soportados por los teléfonos móviles se realizó de forma paulatina, en primer lugar en los servicios más críticos y, por último, en todos los servicios.

Actualmente, casi la totalidad de los teléfonos móviles existentes en el mercado incorporan el modo 3 de seguridad a nivel de enlace en todos los servicios soportados y están protegidos frente a las vulnerabilidades que afectaban a los primeros modelos.

Históricamente, en el análisis de la seguridad de los teléfonos móviles Bluetooth resulta necesario diferenciar los ataques a los primeros modelos, los cuales ya se encuentran hoy en el mercado; y los ataques a teléfonos móviles actuales.

4.1.1 – Ataques a los primeros modelos de teléfonos móviles Bluetooth

Las vulnerabilidades en los primeros modelos se debían a la escasa implementación de los mecanismos de seguridad Bluetooth: autenticación y autorización. Como consecuencia, un atacante podía conectarse a los perfiles soportados por el teléfono móvil utilizando su PC sin necesidad de que los dos dispositivos estuvieran emparejados o el PC estuviera incluido en la lista de dispositivos de confianza del teléfono móvil.

Es posible emular este tipo de ataques en teléfonos móviles actuales, pero se necesita que el dispositivo atacante esté autenticado y autorizado en el teléfono móvil objetivo, por lo que el acto deja de ser un ataque propiamente dicho, ya que requiere la colaboración del usuario víctima.

4.1.1.1 – Bluesnarf (Marcel Holtmann & Adam Laurie, 2003)

El ataque ***Bluesnarf*** se basa en la extracción de archivos de un teléfono móvil Bluetooth a través del Perfil de Carga de Objetos (OBEX Object Push) sin autorización del usuario propietario.

El Perfil de Carga de Objetos (OPUSH u OPP, Object Push Profile) define los requisitos de aplicación para implementar el modelo de uso de *carga de objetos* a través del protocolo OBEX Object Push, el cual ofrece la capacidad de carga y descarga de objetos de datos entre dispositivos Bluetooth.

El propósito inicial del Perfil de Carga de Objetos (OBEX Object Push) era la carga y descarga objetos tales como citas (*vCalendar*) y el intercambio de tarjetas de visita (*vCard*) con otro dispositivo Bluetooth. Actualmente, el perfil conserva esta funcionalidad, aunque también se utiliza para transferencia rápida de archivos.

La vulnerabilidad ***Bluesnarf*** se basa en la implementación incorrecta en los primeros modelos de teléfonos móviles Bluetooth del Perfil de Carga de Objetos (OBEX Object Push), que carecía de mecanismos de autenticación y autorización, y que permitía descargarse mediante una operación *OBEX GET* archivos de nombre conocido, como la agenda de contactos almacenada en el terminal en *telecom/pb.vcf* o el calendario de citas almacenado en *telecom/cal.vs*.

Especificación IrMC de los archivos OBEX:

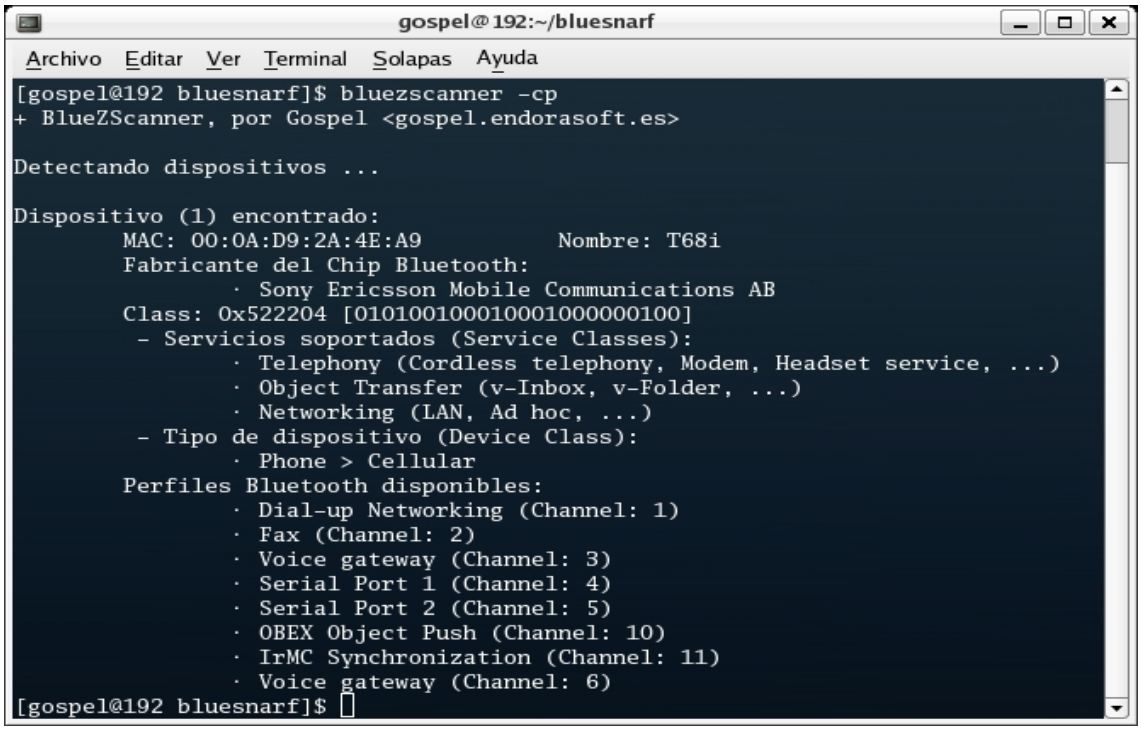
Filename	Description	Supported operations
Device Info		
telecom/devinfo.txt	Information hardware version, software version, serial number, etc. Character sets	GET
telecom/rtc.txt	The Real Time Clock Object contains the current date and time of the device	GET/PUT
Phone Book		
telecom/pb.vcf	Level 2 access (Access entire phonebook database)	GET/PUT
telecom/pb/luid/.vcf	Add new entry	PUT
telecom/pb/0.vcf	Own business card	GET/PUT
telecom/pb/info.log	Supported properties and memory info	GET
telecom/pb/luid/ ###.log	Change log	GET
telecom/pb/luid/ cc.log	Change counter	GET
Calendar		
telecom/cal.vcs	Level 2 access	GET/PUT
telecom/cal/luid/.vcs	Add new entry	PUT
telecom/cal/info.log	Supported properties and memory info	GET

Fuente: Sony-Ericsson AT Commands Online Referente (Developer Guidelines, Octubre 2004)

Hoy en día, la mayoría de teléfonos móviles Bluetooth incorporan únicamente mecanismos de autorización en el acceso al Perfil de Carga de Objetos (OBEX Object Push). Esto significa que el dispositivo remoto debe estar incluido en la lista de dispositivos de confianza del teléfono móvil. En cualquier otro caso, la conexión requerirá confirmación explícita por parte del usuario propietario del teléfono móvil.

A continuación se muestran el procedimiento para llevar a cabo un ataque ***Bluesnarf*** sobre un teléfono móvil Sony-Ericsson™ T68 desde Linux.

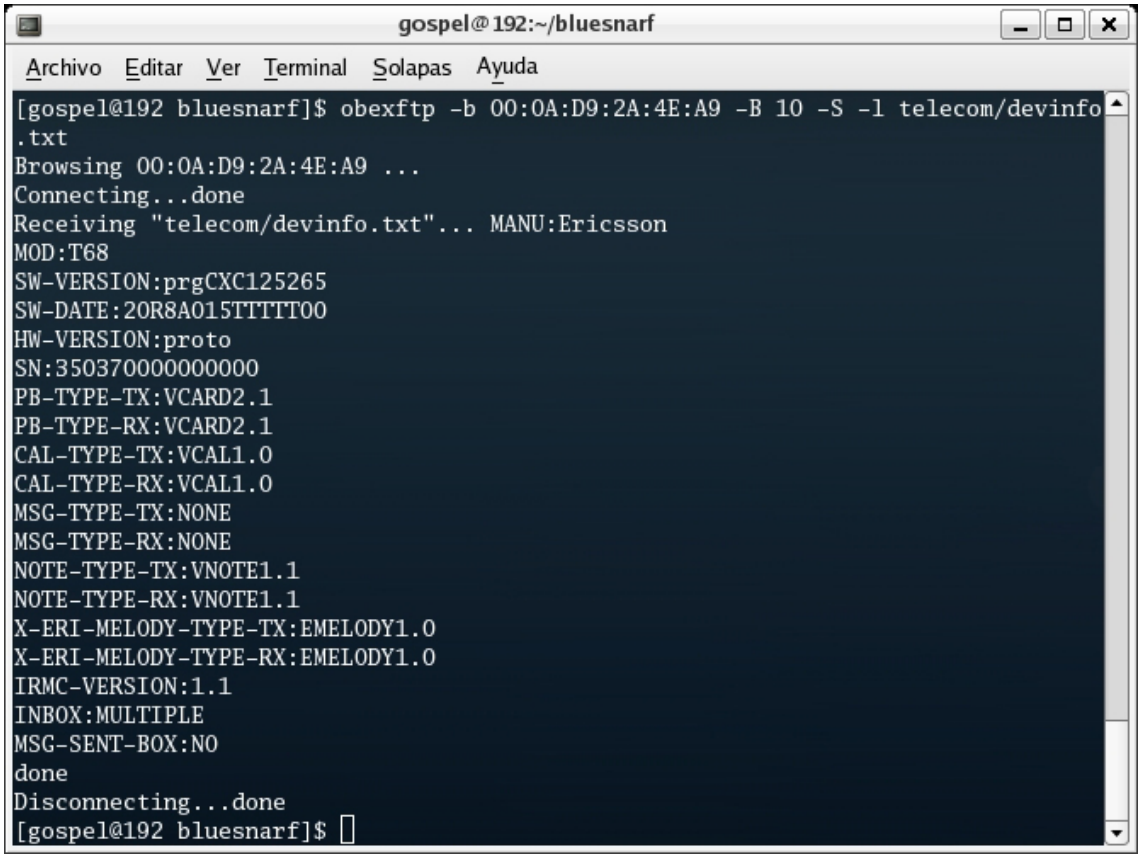
1) Detectar el teléfono móvil Bluetooth objetivo y enumerar los perfiles que soporta.



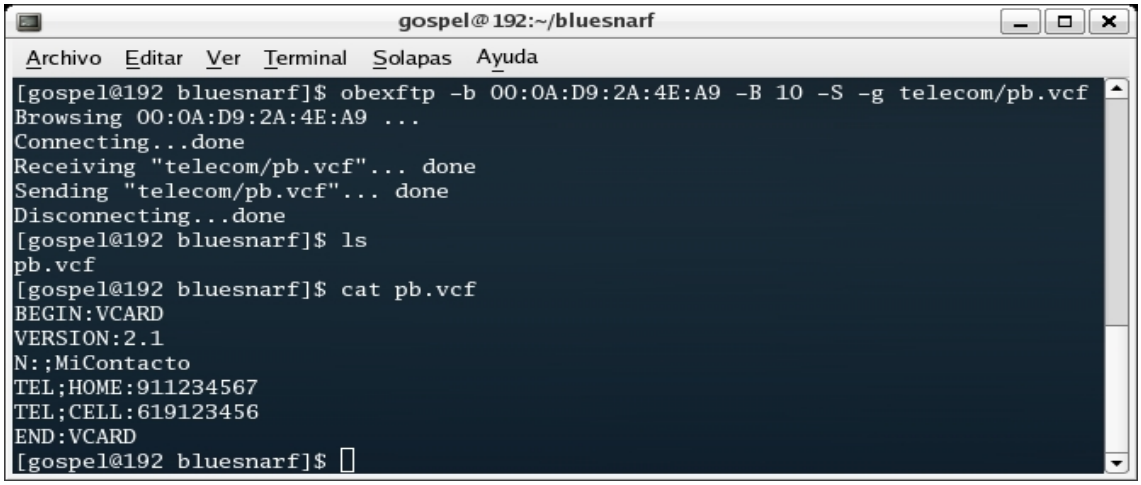
2) Dado que los primeros modelos de teléfonos móviles Bluetooth no implementaban mecanismos de autenticación y autorización en el acceso al Perfil de Carga de Objetos (OBEX Object Push), un atacante podía extraer de forma transparente cualquier archivo disponible en el sistema de ficheros del terminal con ayuda de cualquier aplicación que permitiera realizar operaciones *OBEX GET*, como por ejemplo ***Obexftp***.

Las siguientes capturas corresponden a operaciones *OBEX GET* para extraer archivos del teléfono móvil y visualizarlos a continuación:

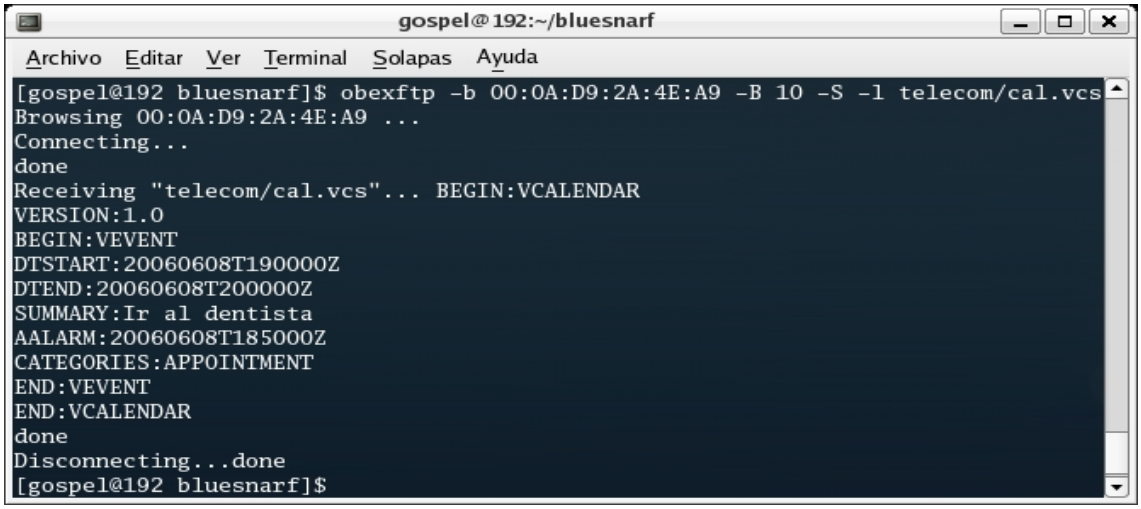
- Lectura del archivo *devinfo.txt*, que contiene información sobre el hardware, software y disponibilidad de recursos del teléfono móvil.



- Extracción y lectura del archivo *pb.vcf*, que contiene la agenda de contactos almacenados en el terminal.



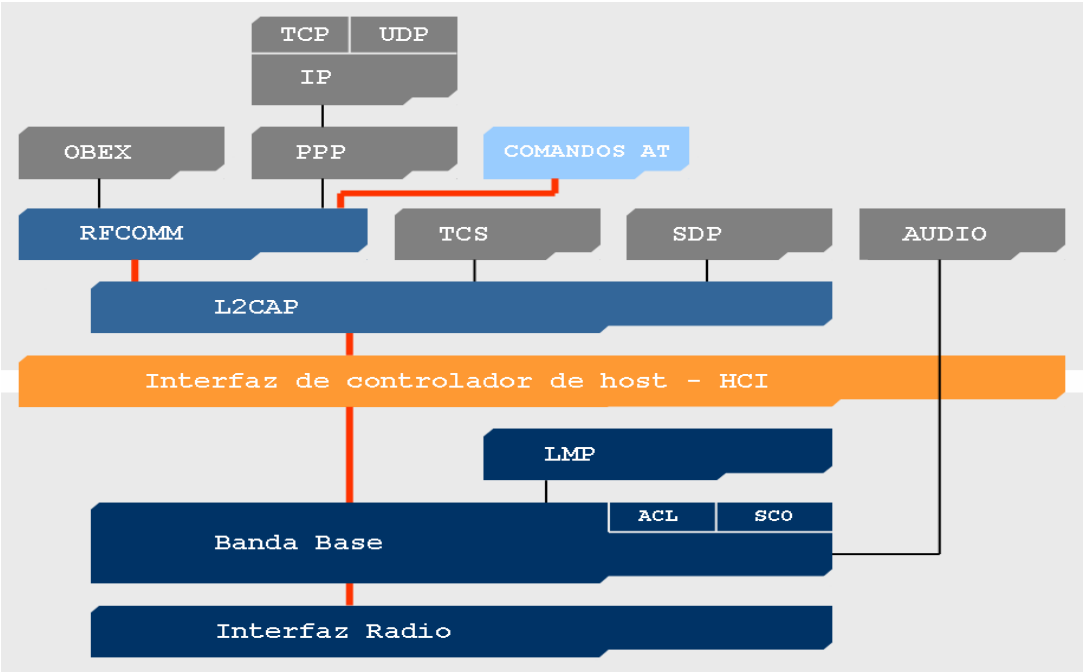
- Lectura del archivo *cal.vcs*, que contiene el calendario de citas.



4.1.1.2 – Bluebug (Martin Herfurt, 2004)

Bluebug es una vulnerabilidad que permite a un atacante establecer una conexión RFCOMM a un canal oculto (no accesible por SDP) sin necesidad de autenticación y ejecutar comandos AT en el terminal.

Como puede apreciarse en el esquema de la pila de protocolos Bluetooth, desde el nivel RFCOMM se puede acceder a la capa de comandos AT. Esto significa que estableciendo una conexión RFCOMM a un determinado canal, el atacante podría iniciar una sesión de comandos AT con el teléfono móvil.



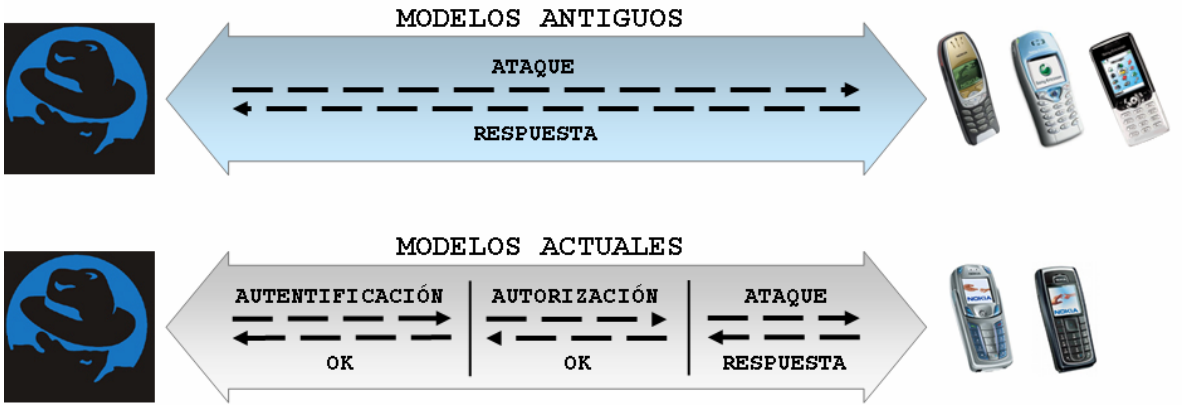
La posibilidad de ejecutar comandos AT en el teléfono móvil, permitiría a un atacante llevar a cabo las siguientes acciones en el terminal comprometido:

- Obtener información básica: Marca, modelo, IMEI,…
- Realizar llamadas de voz, desvío de llamadas,…
- Gestión de la agenda de contactos: Leer, escribir, borrar,…
- Acceso a la agenda de llamadas: Últimas llamadas perdidas, recibidas o realizadas.
- Gestión de mensajes SMS: Leer, escribir y enviar, borrar,…

Bluebug es, sin duda, una de las vulnerabilidades más peligrosas y con mayor impacto en usuarios de teléfonos móviles Bluetooth, no sólo por la violación de privacidad que puede suponer el acceso ajeno a la agenda de contactos o a mensajes SMS, sino por las consecuencias económicas que conlleva la capacidad de efectuar llamadas telefónicas.

Una de las posibles soluciones a esta vulnerabilidad podría ser restringir el acceso a los comandos AT desde el interfaz Bluetooth. Sin embargo, los propios fabricantes desarrollan aplicaciones para sincronizar un equipo PC con la agenda de contactos y la bandeja de entrada de mensajes SMS de los teléfonos móviles. Un ejemplo de aplicación de este tipo es *Nokia™ PC Suite*. Así mismo, algunos dispositivos Bluetooth, como los Manos Libres, requieren el control del teléfono móvil para poder colgar, descolgar e iniciar llamadas telefónicas, para lo cual requieren el uso de los comandos AT. La solución adoptada por los fabricantes para proteger la vulnerabilidad ***Bluebug*** en los teléfonos móviles consiste en añadir mecanismos de autenticación y autorización antes de permitir el establecimiento de una conexión RFCOMM. De esta forma, se necesita la intervención del usuario propietario del teléfono móvil y resulta imposible llevar a cabo un ataque de forma transparente.

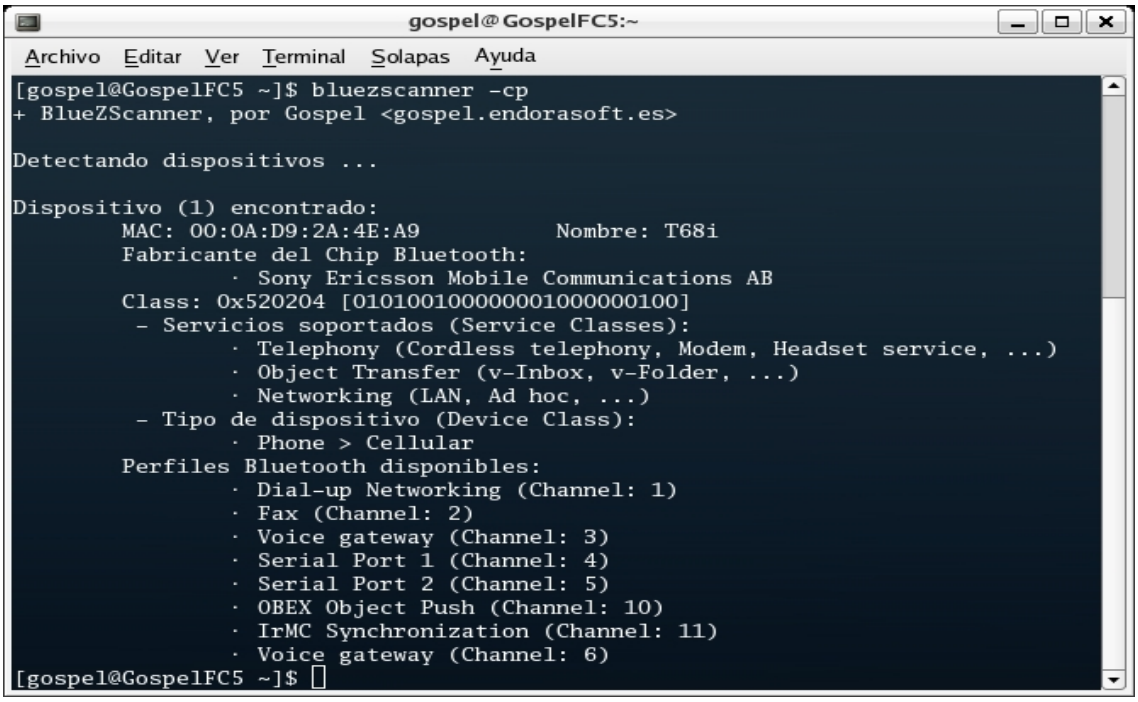
Puesto que los teléfonos móviles actuales también implementan el juego de comandos AT, es posible emular el ataque ***Bluebug*** estableciendo una conexión RFCOMM al teléfono móvil e iniciando una sesión de comandos AT.



A continuación se muestran dos procedimientos para llevar a cabo un ataque ***Bluebug*** sobre un teléfono móvil Sony-Ericsson™ T68. Se llevan a cabo dos ejemplos, uno desde Linux y otro desde Microsoft Windows™ XP SP2.

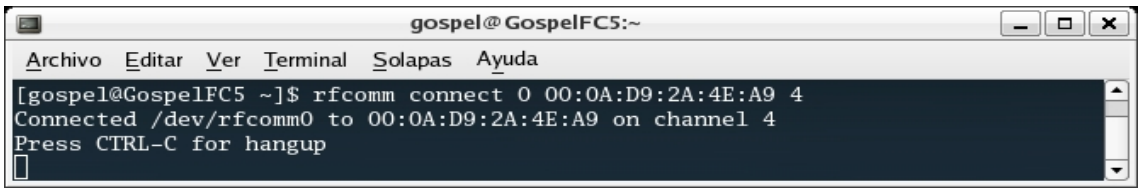
4.1.1.2.1 – Bluebug desde Linux

1) Detectar el teléfono móvil objetivo y enumerar los perfiles que soporta.



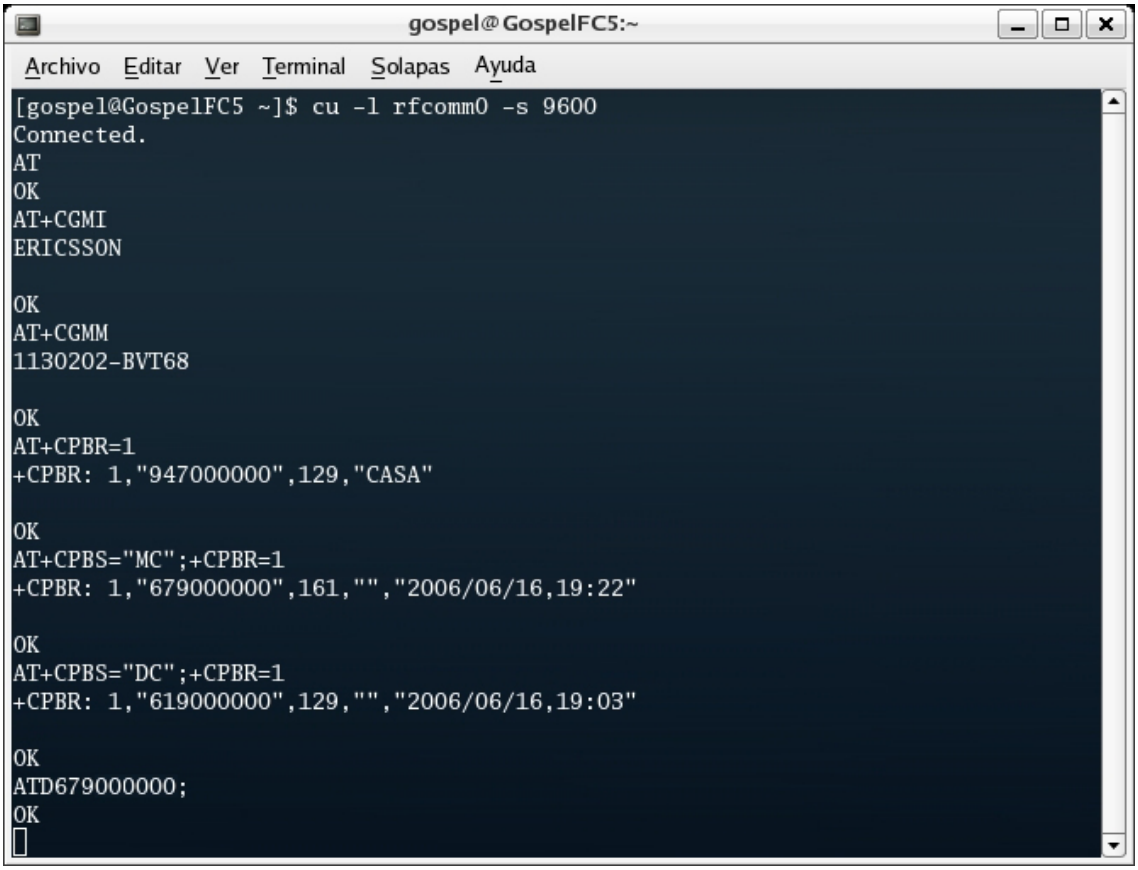
Se puede observar que el canal asociado al Perfil de Puerto Serie es el número 4.

2) El siguiente paso es establecer una conexión RFCOMM al canal 4.



3) Desde otra ventana de shell, hay que lanzar la herramienta ***cu***, que está integrada en el paquete *Taylor UUCP* y puede obtenerse en la siguiente dirección: http://www.airs.com/ian/uucp-doc/uucp.html

Cu permite conectar con el teléfono móvil a través del interfaz rfcomm0 ya establecido e iniciar una sesión de comandos AT. Es necesario especificar la velocidad (en bits por segundo) que utilizará la conexión.

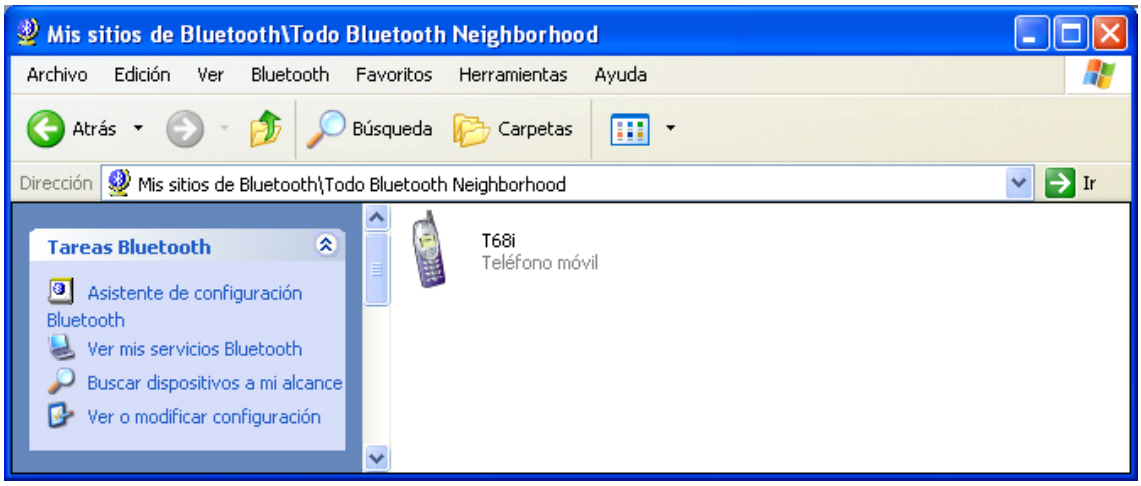


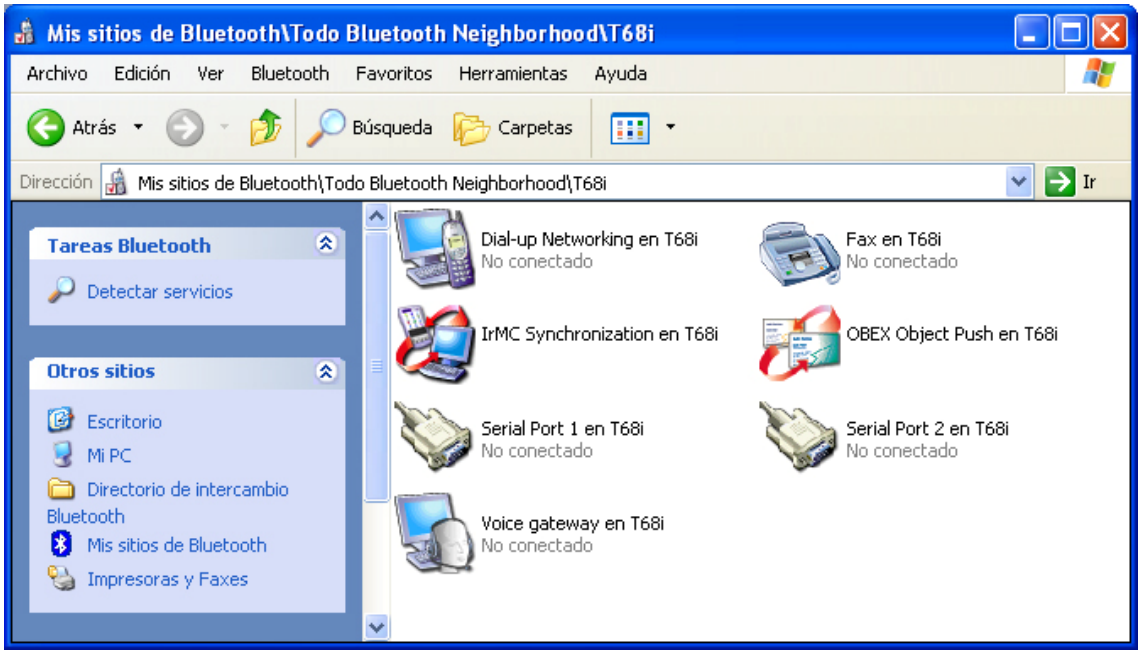
Como muestra la captura, se han ejecutado los siguientes comandos AT en el terminal Sony-Ericsson™ T68:

- *AT*: Comando unitario para comprobar la comunicación.
- *AT+CGMI*: Petición de identificación del fabricante (marca).
- *AT+CGMM*: Petición de identificación del modelo de teléfono.
- *AT+CPBR=1*: Leer la entrada 1 de la agenda de contactos.
- *AT+CPBS="MC";+CPBR=1*: Seleccionar el dispositivo de memoria de llamadas perdidas (MC, *Missed Call List*) y leer la entrada 1.
- *AT+CPBS="DC";+CPBR=1*: Seleccionar el dispositivo de memoria de llamadas realizadas (DC, *Dialled Call List*) y leer la entrada 1.
- *ATD679000000;* :Realizar una llamada de voz a un número.

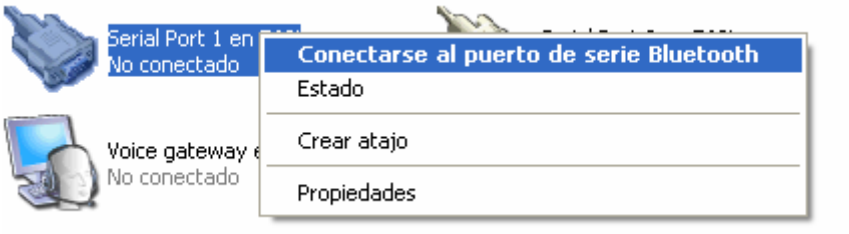
4.1.1.2.2 – Bluebug desde Microsoft Windows™

1) Detectar el teléfono móvil Bluetooth objetivo y enumerar los perfiles que soporta.

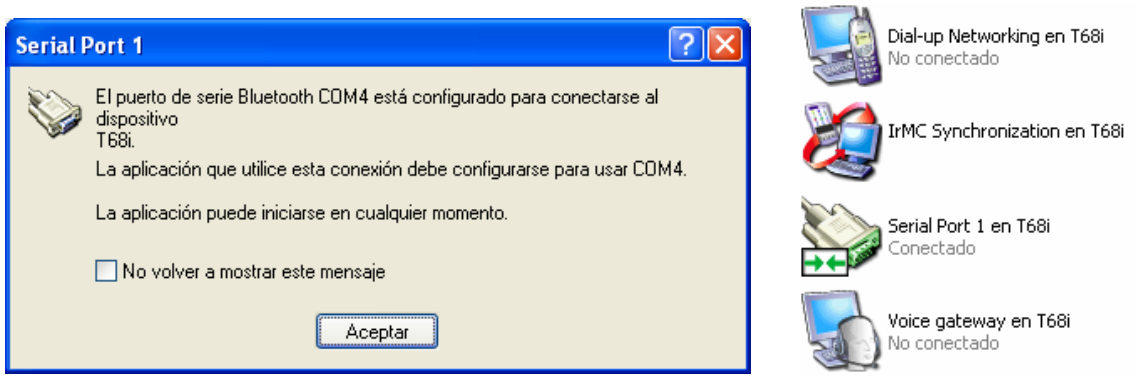




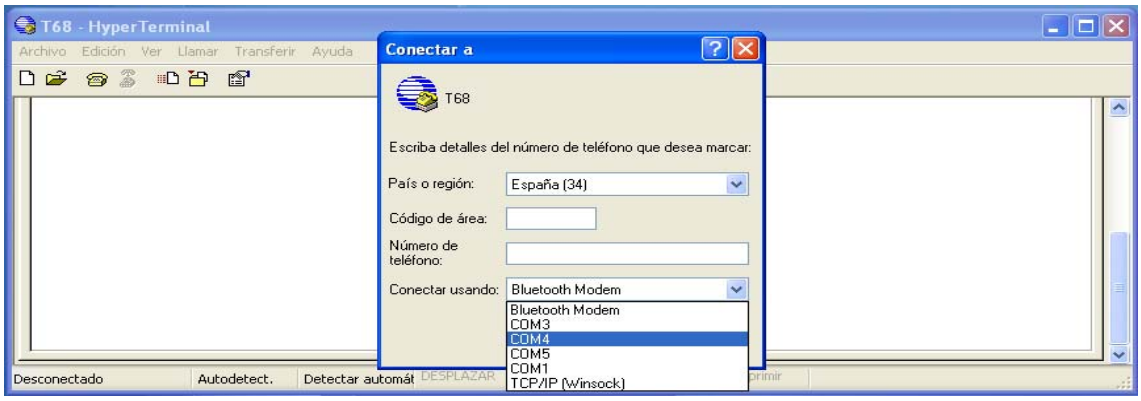
2) Establecer una conexión con el Perfil de Puerto Serie 1

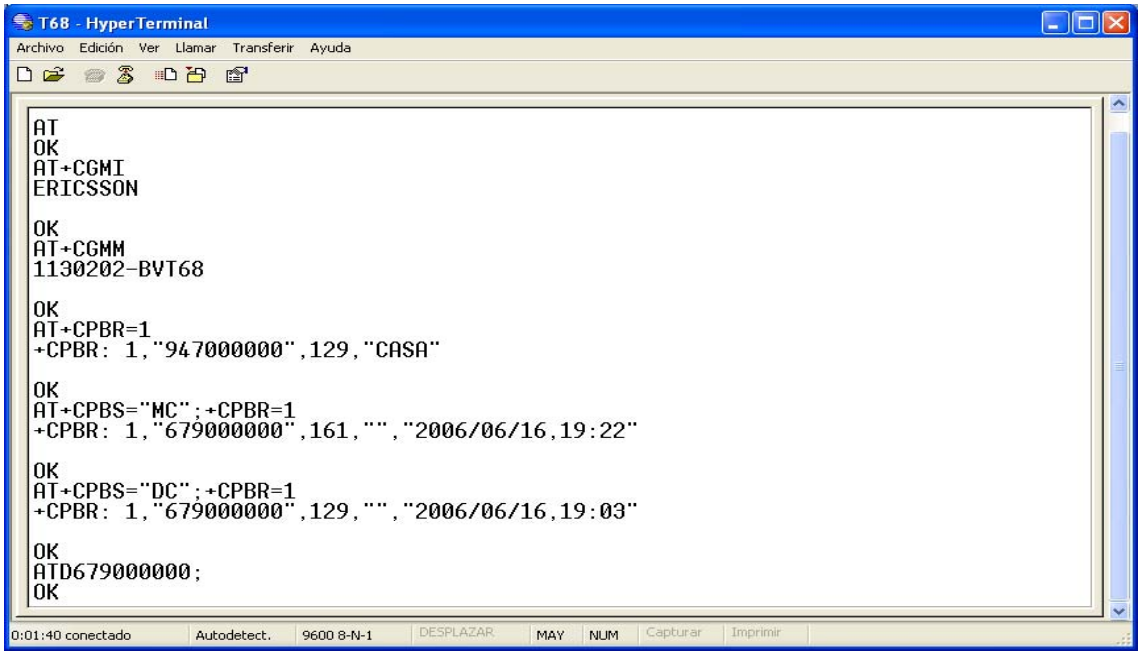


El siguiente mensaje advierte que cualquier aplicación de Windows™ puede acceder a la conexión establecida con el teléfono móvil utilizando el puerto COM4 del PC.



3) Una vez conocido el puerto que permite al PC acceder a la conexión RFCOMM establecida con el teléfono móvil, ***Hyperterminal*** permite crear una nueva conexión a través del puerto COM4.





4.1.1.3 – HeloMoto (Adam Laurie, 2004)

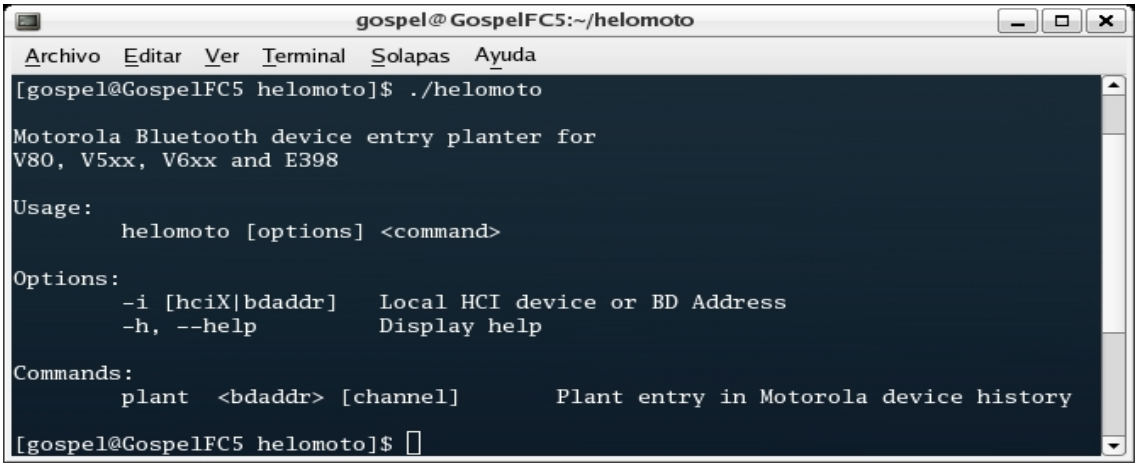
El ataque ***HeloMoto*** es una combinación de los ataque ***Bluebug*** y ***Bluesnarf***. El ataque se denomina ***HeloMoto*** porque afecta a teléfonos móviles Motorola™.

La vulnerabilidad que explota ***HeloMoto*** se basa una implementación incorrecta de la gestión de la lista de dispositivos de confianza en los siguientes modelos Motorola™: V80, v500 y v600.

El ataque se desarrolla del siguiente modo: El atacante inicia una conexión al Perfil de Carga de Objetos (OBEX Push Object) con la intención de enviar una tarjeta de visita o *vCard*. De forma automática y sin necesidad de interacción por parte del usuario propietario del teléfono móvil, el dispositivo atacante es añadido a la lista de dispositivos de confianza del terminal, aunque el proceso de envío haya sido interrumpido por el atacante antes de llegar a su fin. Con el dispositivo incluido en la lista de dispositivos de confianza del teléfono móvil, el atacante puede acceder a perfiles que requieran autorización pero no autenticación, como el caso del Perfil de Auriculares (HeadSet Profile).

Una vez establecida la conexión con el Perfil de Auriculares, el atacante puede acceder a la ejecución de comandos AT en el teléfono móvil comprometido.

Para poder llevar a cabo el ataque ***HeloMoto***, existe una herramienta desarrollada en BlueZ que se puede obtener en la siguiente dirección:
http://trifinite.org/Downloads/helomoto.tgz



4.1.2 – Ataques a teléfonos móviles Bluetooth actuales

Los ataques a los primeros modelos de teléfonos móviles Bluetooth se debían a la pobre implementación de los mecanismos de seguridad, autenticación y autorización que realizaban los fabricantes en sus dispositivos. Se informó a los fabricantes de las vulnerabilidades descubiertas antes de ser publicadas *in the wild*, para evitar graves consecuencias en los usuarios, puesto que casi todos los teléfonos móviles en el mercado eran vulnerables y no existía la posibilidad de corregir el *firmware* del dispositivo.

Los fabricantes tomaron nota de las vulnerabilidades reportadas y las sucesivas unidades comercializadas ya incorporaban mecanismos de seguridad que protegían a esos modelos de los ataques publicados. Una vez informados los fabricantes y tras un periodo de tiempo suficiente para que éstos corrigiesen el fallo en las nuevas versiones de los modelos vulnerables, las vulnerabilidades se publicaban en la red junto con herramientas que explotaban ataques *prueba de concepto*.

Actualmente, menos del 5% de los teléfonos móviles Bluetooth en el mercado son vulnerables a los ataques ***Bluebug***, ***Bluesnarf*** o ***HeloMoto***. Ya se han explorado casi todas las posibles fallas en la implementación de Bluetooth por parte de los fabricantes en sus teléfonos móviles y no se han publicado nuevas vulnerabilidades importantes.

Tras dar por supuesto que los teléfonos móviles actuales incorporan mecanismos de autenticación y autorización para acceder a los perfiles, la tendencia actual es intentar saltarse esas medidas de seguridad implementadas. De esta forma, surgen nuevas modalidades de ataques más complejos y combinados que, aunque en teoría funcionan, en la práctica todavía tienen una escasa probabilidad de poder ser desarrolladas con éxito.

El objetivo sigue siendo el mismo: acceder a la capa de comandos AT del terminal comprometido.

4.1.2.1 – Blueline Attack (Kevin Finisterre, 2006)

El ataque ***Blueline*** consiste en una variación del ataque ***HeloMoto***.

Como se ha descrito anteriormente, la vulnerabilidad que explota ***HeloMoto*** se basa una implementación incorrecta de la gestión de la lista de dispositivos de confianza en algunos teléfonos móviles Motorola™. El ataque ***HeloMoto*** se lleva a cabo enviando una tarjeta de visita o *vCard* a través del Perfil de Carga de Objectos (OBEX Object Push). De forma automática el teléfono móvil agrega al dispositivo a su lista de dispositivos de confianza, de forma que el atacante queda acreditado para acceder a perfiles que requieran autorización, pero no autenticación, como el Perfil de Auriculares (Headset Profile). Una vez establecida una conexión con este perfil, el atacante tiene acceso a la ejecución de comandos AT en terminal.

Los nuevos modelos de Motorola™, como PEBL, no son vulnerables al ataque ***HeloMoto*** y requieren que el usuario confirme explícitamente cualquier conexión a un perfil que requiera autorización si el dispositivo no está agregado a la lista de dispositivos de confianza del teléfono móvil.

Uno de los perfiles del Motorola™ PEBL que requiere autorización, que no autenticación, es el Perfil de Pasarela de Voz (Voice Gateway Profile), asociado al canal 3. El *Service Record* correspondiente es:

```
Service Name: Voice Gateway
Service Description: Headset Audio Gateway
Service Provider: T-Mobile
Service RecHandle: 0x10003
Service Class ID List:
 "Headset Audio Gateway" (0x1112)
 "Generic Audio" (0x1203)
Protocol Descriptor List:
 "L2CAP" (0x0100)
 "RFCOMM" (0x0003)
   Channel: 3
```

Cualquier intento de conexión a este perfil, requerirá una confirmación explícita por parte del usuario para autorizar la conexión entrante. Se mostrará por defecto el siguiente mensaje en la pantalla del teléfono móvil:

Fuente: http://www.digitalmunition.com/

La cuestión reside en persuadir al usuario del teléfono móvil para que confirme la autorización de conexión al perfil. No obstante, si el nombre del dispositivo resulta sospechoso o, simplemente por conducta preventiva, el usuario habitualmente denegará la conexión.

El ataque ***Blueline*** permite provocar una falsificación o *spoofing* del interfaz sustituyendo el mensaje original de la ventana de aviso de conexión entrante por cualquier texto deseado con sólo modificar el nombre del dispositivo atacante por una cadena de caracteres maliciosa. Esta cadena de caracteres puede contener caracteres *0x0d* para el salto de línea.

De esta forma, la ejecución de los siguientes comandos provocará que el mensaje de aviso original sea sustituido por otro mensaje malicioso que podría engañar a cualquier usuario de teléfono móvil y forzarle a que acepte la conexión, en cuyo caso, de forma automática, el dispositivo atacante quedaría agregado a la lista de dispositivos de confianza del teléfono móvil comprometido.

```
# hciconfig hci0 name `perl -e 'print
"Press\x0dgrant\x0dto\x0ddisable\x0dmute\x0d\x0d"'`
# rfcomm connect 0 00:15:A8:74:87:3E 3   (espera a que usuario acepte)
Connected /dev/rfcomm0 to 00:15:A8:74:87:3E on channel 3
Press CTRL-C for hangup
```

El texto malicioso se traduce por: "*Pulse aceptar para deshabilitar la función de silencio*".

En caso de que el usuario atacado sea engañado y confíe en el aviso mostrado por pantalla, el dispositivo atacante dispondrá de autorización para acceder al Perfil de Pasarela de Audio y ejecutar comandos AT en el terminal comprometido.

Fuente: http://www.digitalmunition.com/

4.1.2.2 – Blue MAC Spoofing (Kevin Finisterre, 2005 - Bluehack, 2006)

La especificación Bluetooth describe que la autorización es el procedimiento que determina los derechos que tiene un dispositivo Bluetooth para acceder a los servicios que ofrece un sistema. El mecanismo de autorización en dispositivos Bluetooth se lleva a cabo mediante *niveles de confianza*. Si un dispositivo está incluido en la lista de dispositivos del sistema, entonces está autorizado para acceder a cualquiera de los servicios que necesite autorización.

La especificación Bluetooth describe que la autenticación es el procedimiento por el cual un dispositivo Bluetooth verifica su identidad en otro dispositivo para poder acceder a los servicios que ofrece. Todas las funciones de seguridad de nivel de enlace están basadas en el concepto de claves de enlace, las cuales se generan durante la etapa de emparejamiento y posteriormente se almacenan individualmente en cada par de dispositivos. La autenticación no requiere la intervención del usuario; implica un esquema de desafío/respuesta entre cada par de dispositivos que emplea una clave de enlace secreta común K_{ab} de 128 bits. Consecuentemente, este esquema se utiliza para autentificar dispositivos, no usuarios.

El proceso de emparejamiento, basado en el esquema desafío/respuesta, transcurre como se describe a continuación:

7) El dispositivo reclamante envía su dirección BD_ADDR al dispositivo verificador.

8) El verificador devuelve un desafío aleatorio de 128 bits al demandante.

9) El reclamante usa el algoritmo E1 para generar la respuesta de autenticación (SRES) de 32 bits, usando como parámetros de entrada la dirección BD_ADDR del reclamante, la clave de enlace K_{ab} almacenada y el desafío. El verificador realiza la misma operación en paralelo.

10) El reclamante devuelve la respuesta SRES al verificador.

11) El verificador comprueba la respuesta SRES recibida por el reclamante con la respuesta SRES calculada por él.

12) Si los valores de SRES coinciden, el verificador establece la conexión.

El ataque ***Blue MAC Spoofing*** permite a un atacante suplantar la identidad de un dispositivo de confianza para atacar un teléfono móvil y utilizar sus credenciales para acceder a servicios que requieren autenticación y/o autorización.

El ataque ***Blue MAC Spoofing*** se puede desarrollar en dos niveles:

- Suplantación de la dirección MAC de un dispositivo de confianza para acceder a servicios que requieren autorización.

- Suplantación de la dirección MAC de un dispositivo de confianza y obtención de la clave de enlace K_{ab}, generada durante el emparejamiento del dispositivo suplantado con el teléfono móvil, para acceder a servicios que requieren autenticación.

4.1.2.2.1 – Suplantación de dirección MAC

En un primer nivel, el ataque ***Blue MAC Spoofing*** se basa en suplantar la dirección MAC de un dispositivo de confianza para atacar un teléfono móvil y utilizar sus credenciales para acceder a servicios que requieren únicamente autorización. Uno de estos servicios puede ser el Perfil de Carga de Objectos (OBEX Object Push), el cual está implementado en la mayoría de teléfonos móviles sin necesidad de autenticación, sólo se necesita autorización.

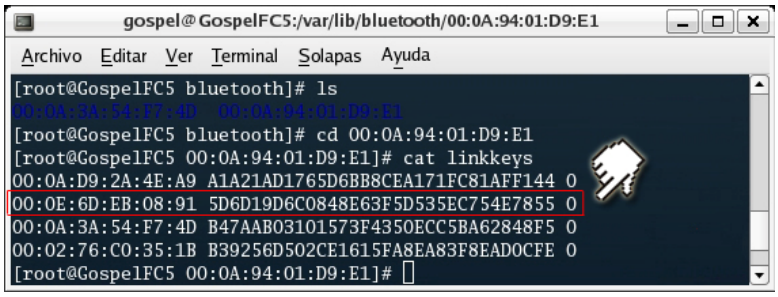
El Perfil de Carga de Objetos permite cargar y descargar objetos de datos entre dispositivos Bluetooth a través del protocolo OBEX Object Push.

Dado que no es posible modificar la dirección MAC en la mayoría de dispositivos Bluetooth existentes en el mercado, únicamente en módulos hardware especiales, para representar la prueba de concepto del ataque se puede suponer que tanto el dispositivo suplantado como el atacante utilizan el mismo dispositivo USB Bluetooth. De esta forma, se puede simular la suplantación de MAC.

En la siguiente demostración participan 3 dispositivos: dos equipos PCs con Linux y un teléfono móvil. Uno de los equipos PC actúa como dispositivo *PC amigo* y el otro como dispositivo *PC atacante*.

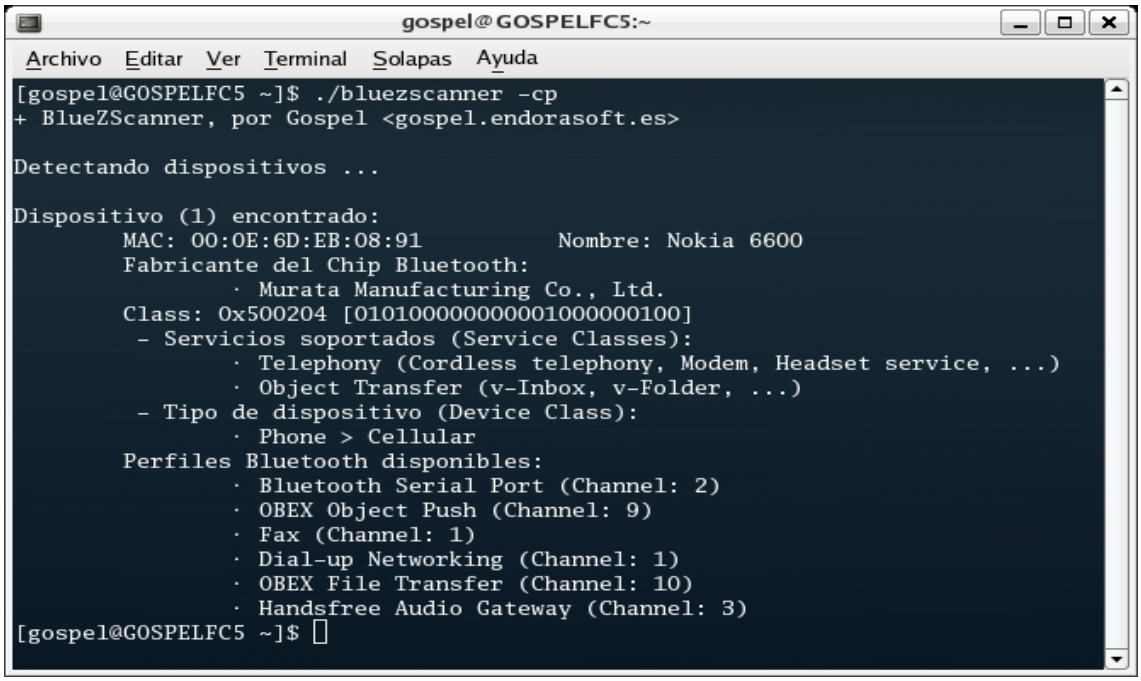


En primer lugar, el teléfono móvil y el *PC amigo* se emparejan. En el teléfono móvil se observa como el *PC amigo* aparece en la lista de dispositivos vinculados. En el sistema Linux del *PC amigo*, queda añadida la clave de enlace K_{ab} generada durante el emparejamiento al archivo /var/lib/bluetooth/BD_ADDR/linkkeys, siendo BD_ADDR la dirección MAC del módulo Bluetooth local.

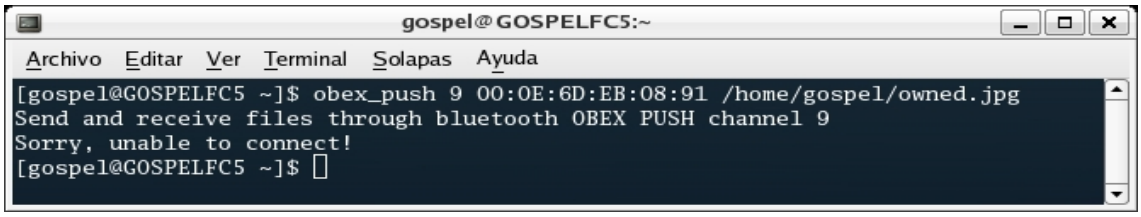


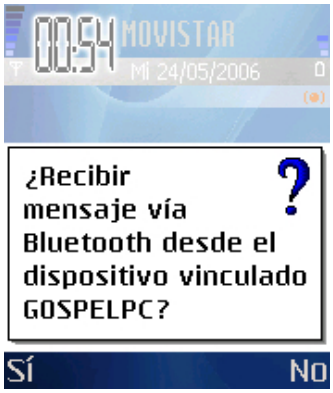
A continuación, se desconecta el dispositivo USB Bluetooth del *PC amigo* y se conecta al *PC atacante*. Con esta acción se simula que el *PC atacante* suplanta la dirección MAC del *PC amigo*.

Desde el *PC atacante* se inicia un escaneo en busca del teléfono móvil y se listan los perfiles que soporta, a fin de conocer el canal asociado al Perfil de Carga de Objetos (OBEX Object Push). En este caso se trata del canal 9.



La herramienta ***ObexPush***, contenida en el paquete *bluez-utils*, permite el intercambio de archivos a través del protocolo OBEX Object Push. Con ayuda de esta herramienta, el atacante puede enviar un archivo malicioso al teléfono móvil objetivo.

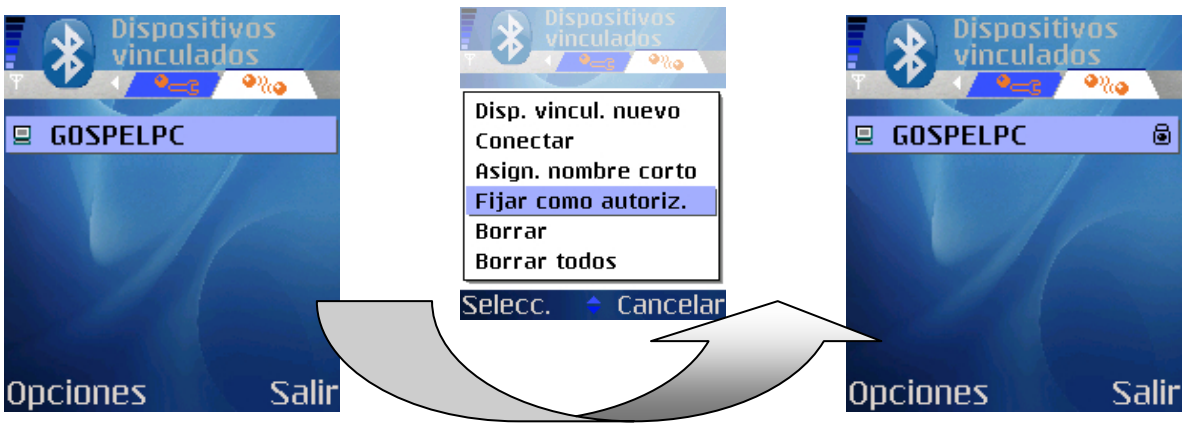




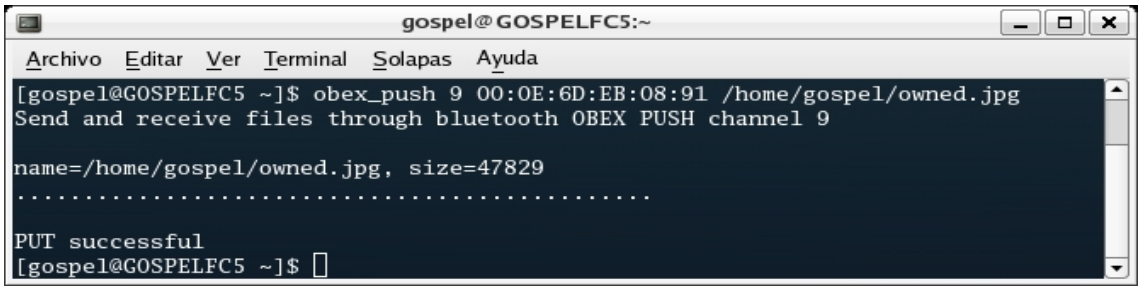
Al tratar de enviar el archivo, en el teléfono móvil aparece una notificación solicitando al usuario la autorización para recibir el archivo enviado por el *PC atacante*, aunque a los ojos del teléfono móvil, el dispositivo origen de la conexión es el *PC amigo*, ya que tiene la misma dirección MAC.

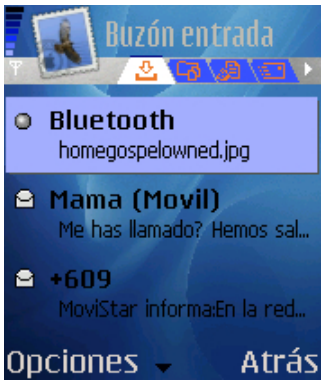
Esto se debe a que, inicialmente, en el teléfono móvil no se había añadido la dirección MAC del *PC amigo* a la lista de dispositivos de confianza explícitamente marcando la opción de omitir notificaciones de autorización en conexiones desde ese dispositivo.

Para que el ataque ***Blue MAC Spoofing*** pueda llevarse a cabo con éxito, es condición necesaria que el dispositivo cuya dirección MAC va a ser suplantada por el atacante para obtener sus credenciales, haya sido incluido en la lista de dispositivos de confianza del teléfono móvil. Esto, en el caso de teléfonos móvil basado en Symbian™ OS, ha de hacerse marcando explícitamente la opción de omitir notificaciones de conexiones provenientes de ese dispositivo.



Una vez que el atacante sepa que el dispositivo *PC amigo* ha sido añadido a la lista de dispositivos de confianza del teléfono móvil, estará en disposición de llevar a cabo el ataque ***Blue MAC Spoofing*** con éxito, ya que se encuentra autorizado para acceder al servicio.





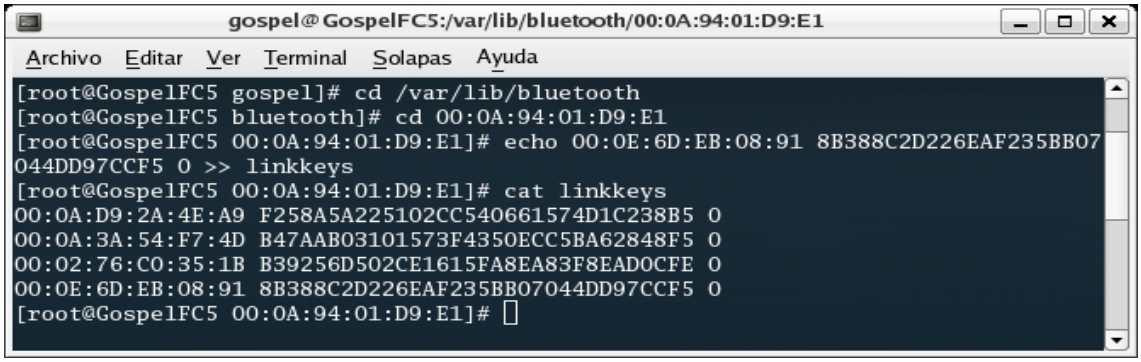
Tras el envío del fichero desde el *PC atacante* con ayuda de la utilidad ***ObexPush***, el teléfono móvil recibe el archivo malicioso sin que se requiera al usuario autorización para recibir el mismo o se le notifique de cualquier conexión Bluetooth entrante. En el caso de Symbian™ OS, el archivo recibido aparece en pantalla con el formato de nuevo mensaje SMS recibido y al abrirlo, se muestra la imagen adjunta directamente.

4.1.2.2.2 – Suplantación de dirección MAC y robo de clave de enlace

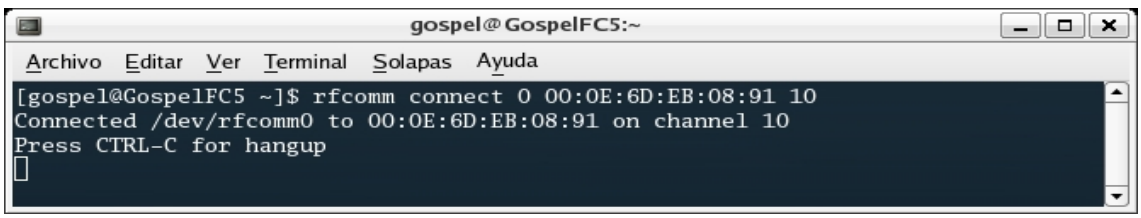
En un nivel más avanzado, el ataque ***Blue MAC Spoofing*** se basa en suplantar la dirección MAC de un dispositivo de confianza para atacar un teléfono móvil junto con la clave de enlace K_{ab} común a ambos dispositivos y utilizar sus credenciales para acceder a servicios que requieren autenticación.

La suplantación de dirección MAC es una acción trivial. La obtención de la clave de enlace K_{ab}, en cambio, puede resultar más compleja de llevar a cabo. Puede implicar el desarrollo de una explotación de bugs en el dispositivo a suplantar para poder acceder a las claves de enlace que almacena, o también llegar al uso de la ingeniería social para engañar al usuario y que se preste a revelar el contenido de un archivo de claves de enlace. En cualquier caso, no es materia de este proyecto entrar en este tipo de consideraciones y se da por supuesto que el atacante puede encontrar la manera de obtener la clave de enlace común al dispositivo de confianza que quiere suplantar y al teléfono móvil objetivo del ataque.

En el caso de que el atacante utilice un equipo Linux, la clave de enlace conseguida se debería almacenar en el archivo /var/lib/bluetooth/BD_ADDR/linkkeys, siendo BD_ADDR la dirección MAC del módulo Bluetooth local.



Una vez que el atacante dispone de la dirección MAC del dispositivo de confianza y de la clave de enlace asociada al teléfono móvil, puede desarrollar el ataque ***Blue MAC Spoofing*** a alto nivel y utilizar las credenciales del dispositivo suplantado para acceder a servicios que requieren autenticación en el teléfono móvil.

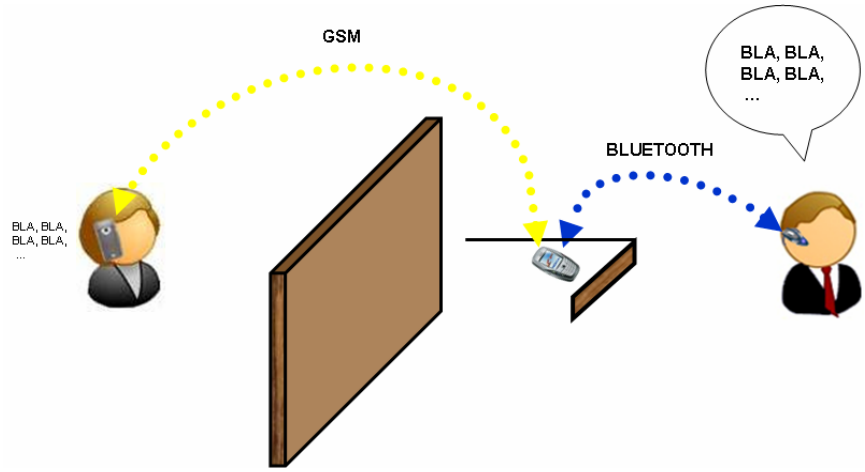


4.2 – Ataques dispositivos Manos Libres

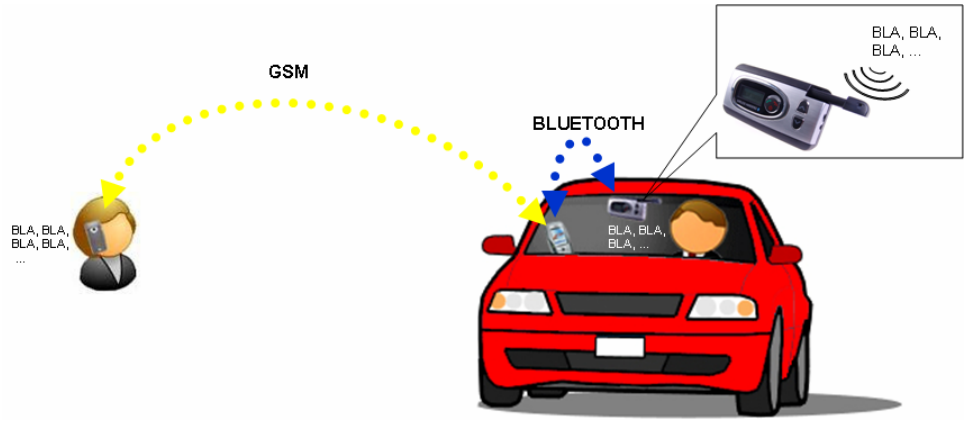
El Perfil de Auriculares (HS, HeadSet Profile) define los protocolos y procedimientos para el modelo de uso que permite utilizar un dispositivo auricular de última generación como interfaz de entrada y salida de audio de otro dispositivo, generalmente un teléfono móvil o un PC; con el propósito de incrementar la libertad de movimiento del usuario al mismo tiempo que se mantiene la confidencialidad de la conversación.

El modelo de uso del Perfil de Auriculares permite multitud de configuraciones y define tres escenarios de uso habituales:

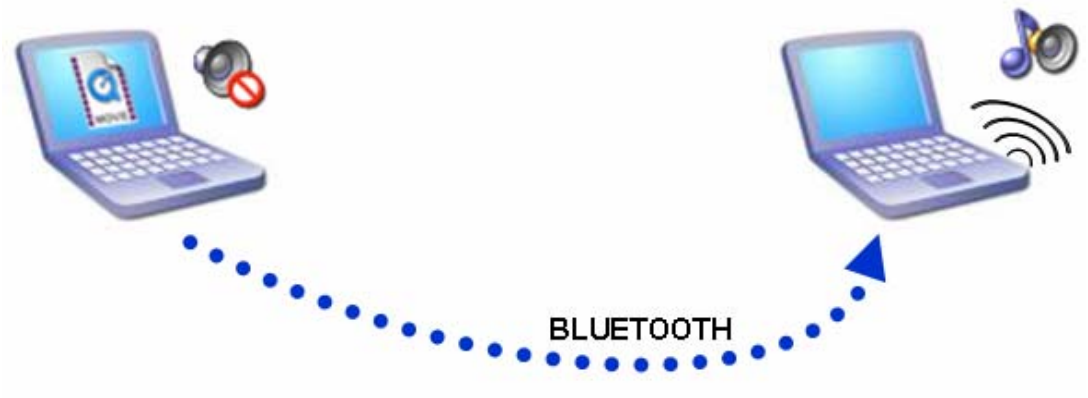
- **Manos Libres Auriculares** (Hands-Free HeadSet) conectado a un teléfono móvil: Permite al usuario mantener conversaciones telefónicas sin necesidad de acercar el terminal al oído.



- **Manos Libres de automóvil** (Hands-Free Car Kit) conectado a un teléfono móvil: Permite al usuario mantener conversaciones telefónicas en el interior de un vehículo sin necesidad de apartar las manos del volante para sostener el teléfono móvil.



- **Pasarela de audio** entre dos dispositivos Bluetooth cualesquiera: Permite a un usuario configurar dos equipos Bluetooth, que no tienen porqué tratarse de auriculares, sino simples PCs o PDAs, y establecer una pasarela de audio entre los dos, de forma que el audio que reproduce el software de un dispositivo, se transmite al otro dispositivo a través del enlace SCO (Synchronous Connection Oriented) y puede ser proyectado por los altavoces del segundo. Así mismo, el audio recogido por el micrófono de un dispositivo se transmite al otro dispositivo, donde puede ser grabado en un archivo de sonido.



El modelo de uso *Manos Libres* es uno de los más significativos de la tecnología Bluetooth y su uso se ha ido extendiendo en los últimos años debido, sobre todo, a la reciente popularidad de los dispositivos Manos Libres de automóvil, los cuales representan el único medio legal de conducir un vehículo al tiempo que se mantiene una conversación telefónica.

A continuación se detallan ataques específicos a los tres modelos de uso habituales del Perfil de Auriculares mencionados. Todos los ataques a dispositivos Manos Libres persiguen los mismos objetivos:

- Capturar el audio recogido por el micrófono del dispositivo, lo cual permitiría escuchar conversaciones privadas.

- Inyectar audio que sería reproducido por los altavoces del dispositivo, lo cual permitiría proyectar mensajes de voz a los usuarios.

En cualquiera de los casos, la consecución con éxito de un ataque a un dispositivo Manos Libres compromete la intimidad del usuario propietario del dispositivo, la violación de la confidencialidad en conversaciones que tengan lugar en las proximidades de un dispositivo comprometido y, en el caso de ataques a dispositivos Manos Libres de automóvil, la seguridad al volante de los pasajeros de un automóvil.

4.2.1 – The Car Whisperer (Martin Herfurt, 2005)

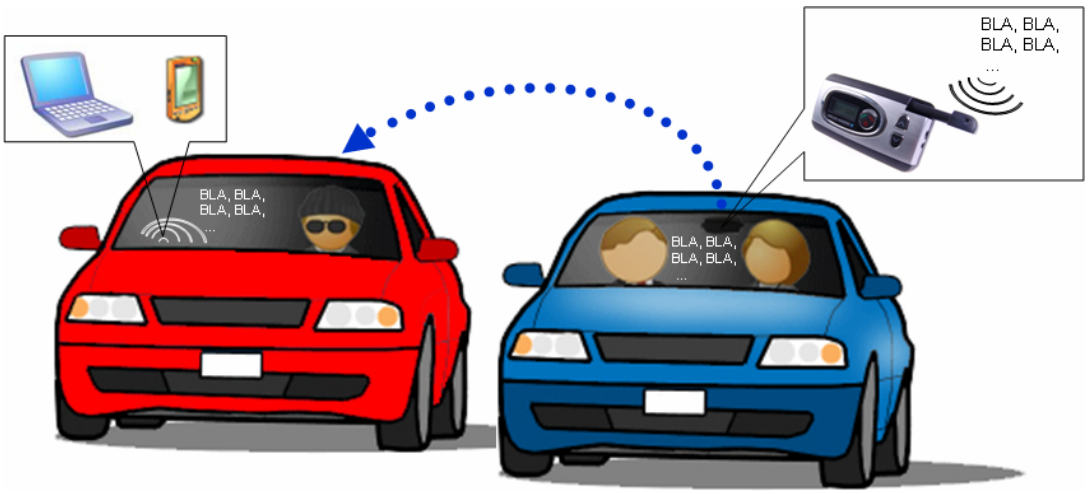
El ataque ***Car Whisperer*** tiene como objetivo los dispositivos Manos Libres de automóvil, aunque también afecta a los dispositivos Manos Libres Auriculares.

El propósito original del proyecto ***Car Whisperer*** era sensibilizar a los fabricantes de dispositivos Manos Libres Bluetooth para automóvil acerca de la amenaza para la seguridad que supone incorporar claves PIN por defecto como medio para emparejarse con estos dispositivos.

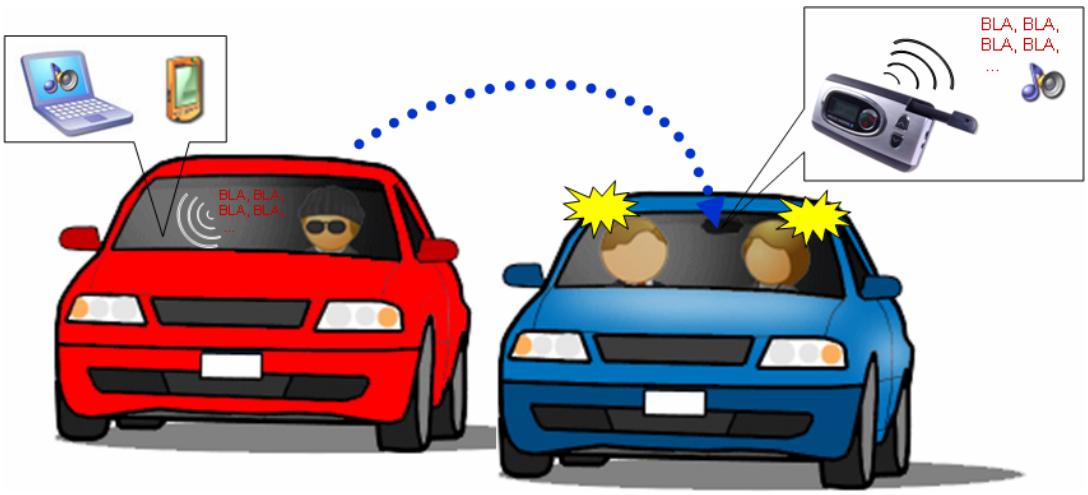
La primera vez que dos dispositivos Bluetooth intentan establecer comunicación, se utiliza un procedimiento de emparejamiento para crear una clave de enlace común a partir de un código de seguridad Bluetooth (clave PIN), que es requerido a cada dispositivo y que debe ser el mismo para los dos. Posteriormente, cuando los mismos dispositivos se comuniquen, utilizarán la clave de enlace para funciones de autenticación y cifrado.

El hecho de incorporar una clave PIN por defecto en un dispositivo Bluetooth significa que cualquier usuario con conocimiento de esa clave estándar puede emparejarse con el dispositivo y comunicarse con él de forma autorizada. En el caso de un Manos Libres, un atacante podría acceder a las funciones de audio implementadas en el terminal y llevar a cabo las siguientes acciones con fines maliciosos:

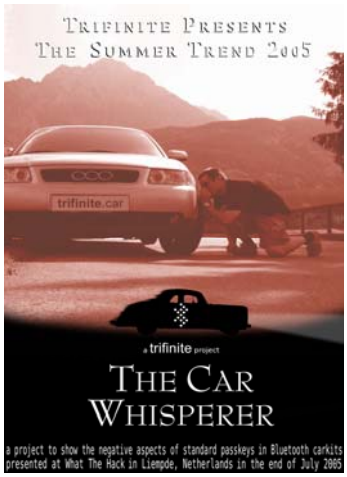
- Capturar el audio recogido por el micrófono del dispositivo, lo cual permitiría escuchar conversaciones privadas en el interior del vehículo.



- Inyectar audio que sería reproducido por los altavoces del dispositivo, lo cual permitiría proyectar mensajes de voz a los ocupantes del vehículo.



El ataque ***Car Whisperer*** descubierto por ***Trifinit Groupe*** se basa en la utilización de una herramienta para Linux con BlueZ. Esta herramienta se puede obtener en la siguiente dirección: http://trifinite.org/trifinite_stuff_carwhisperer.html



La herramienta se compone de varios programas encargados de:

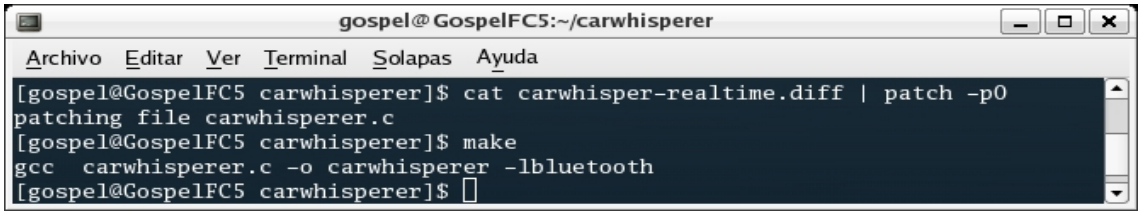
- Identificar dispositivos Manos Libres.
- Utilizar claves PIN estándares para emparejarse con el dispositivo.
- Crear una pasarela de audio estableciendo enlaces SCO para la transmisión de audio en ambas direcciones.

La herramienta ***Car Whisperer*** desarrollada por el grupo ***Trifinite*** permite capturar e inyectar audio en tiempo diferido, es decir, permite capturar audio que es almacenado en un archivo *.raw* para posterior reproducción y también permite inyectar audio pregrabado en mensajes *.raw*.

Con el fin de solucionar el inconveniente de no poder escuchar el audio recogido al mismo tiempo que se almacena, Kevin Finisterre publicó un parche que, aplicado a la herramienta ***Car Whisperer***, permite realizar la operación de captura de audio en tiempo real, es decir, el audio recogido por el micrófono del Manos Libres se reproduce automáticamente por la tarjeta de sonido del PC del atacante al tiempo que se graba en un fichero.

Se puede obtener el parche de Kevin Finisterre en la siguiente dirección: http://www.digitalmunition.com/carwhisper-realtime.tar

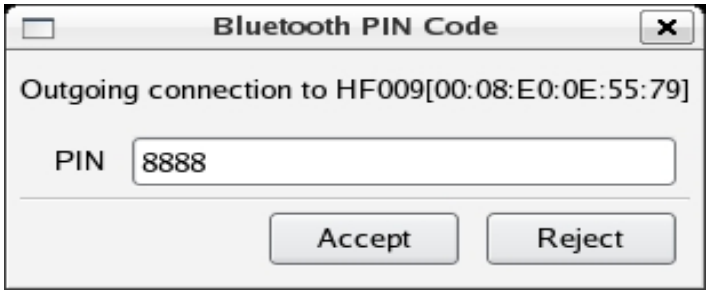
Para aplicar el parche a la herramienta original, basta con descomprimir los archivos del programa ***Car Whisperer*** y del parche en el mismo directorio, aplicar el parche y compilar el código fuente para obtener la herramienta completa.



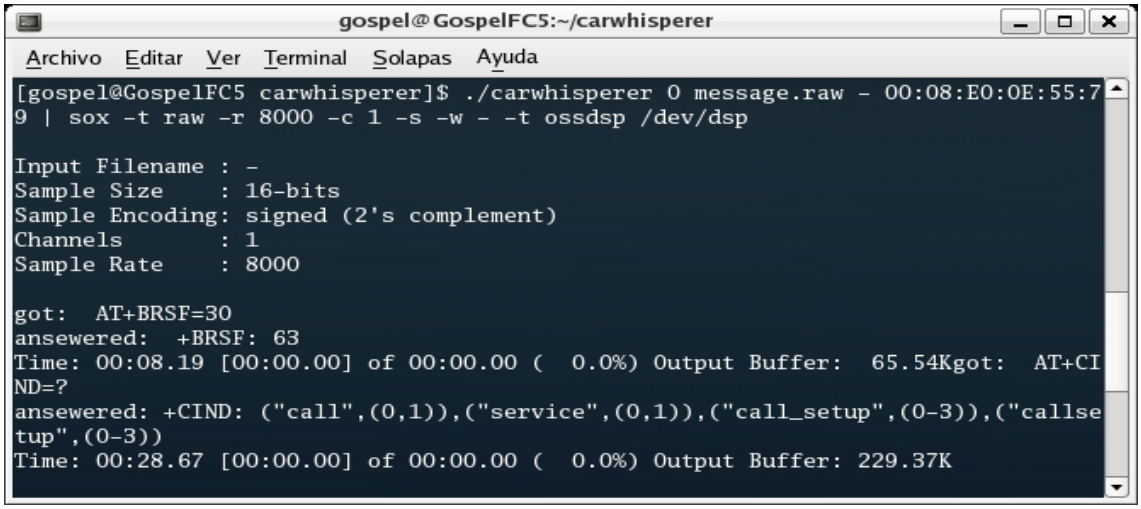
El procedimiento para desarrollar el ataque ***Car Whisperer*** es el siguiente:

En primer lugar, el atacante debe ser capaz de encontrar en sus proximidades un dispositivo Manos Libres Bluetooth en modo visible o *discoverable*. Es condición necesaria que el dispositivo se encuentre activo pero no conectado a ningún otro dispositivo en ese momento.

Un intento de conexión al Perfil de Auriculares lanzará un proceso de emparejamiento que requiere al atacante la especificación de un código de seguridad Bluetooth (clave PIN) que debe coincidir con la clave por defecto almacenada en el dispositivo Manos Libres. El conocimiento de esta clave es trivial, ya que las claves estándar determinadas por los fabricantes suelen ser tan simples como 0000, 1234, 8888…



Si el proceso de emparejamiento se lleva a cabo con éxito, el PC del atacante quedará autorizado para acceder al Perfil de Auriculares y estará en disposición de lanzar la herramienta ***Car Whisperer***.



Tras la ejecución de la herramienta ***Car Whisperer***, el atacante habrá conseguido llevar a cabo las siguientes acciones:

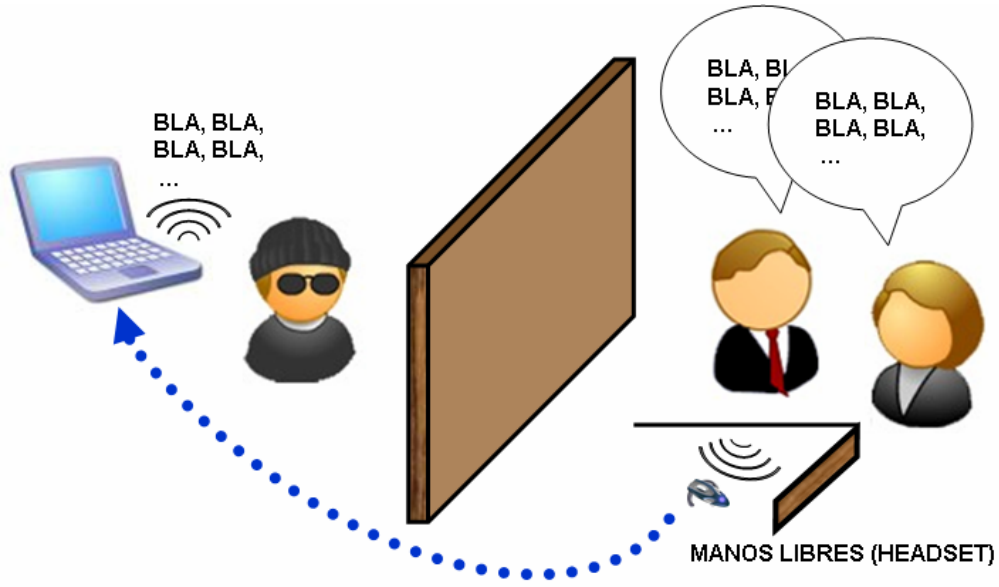
- Inyectar el mensaje de audio *message.raw* que será reproducido por los altavoces del dispositivo Manos Libres en el interior del vehículo.

- Capturar el audio recogido por el micrófono del dispositivo Manos Libres y reproducirlo en tiempo real en la tarjeta de sonido del PC, permitiendo al atacante escuchar conversaciones privadas en el interior del vehículo.

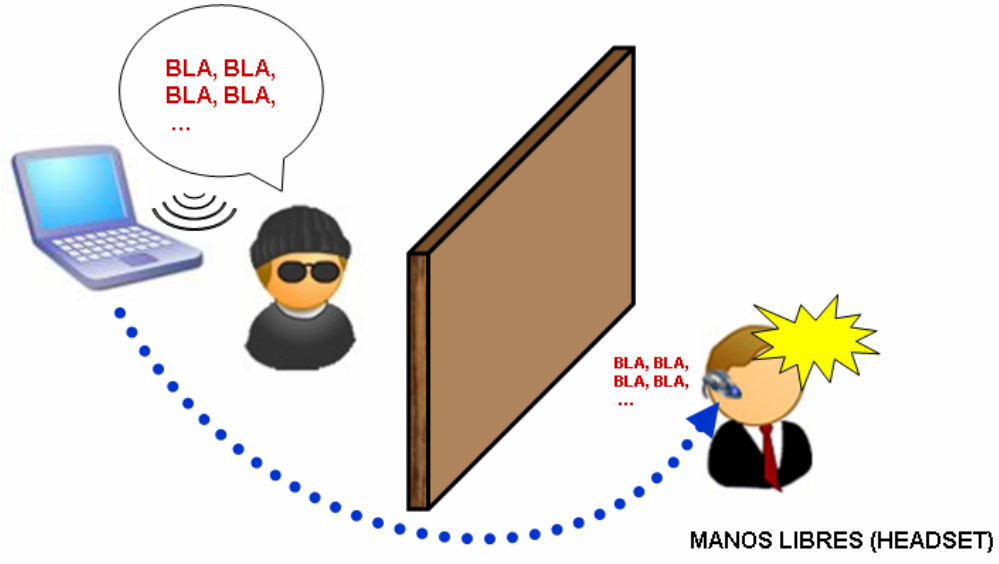
4.2.2 – Headsets Hijacking (Kevin Finisterre, 2005)

El ataque ***Car Whisperer*** también afecta a dispositivos Manos Libres Auriculares, ya que, del mismo modo que los dispositivos Manos Libres de automóvil, incorporan la vulnerabilidad del código de seguridad Bluetooth (clave PIN) por defecto. Cualquier atacante podría emparejarse con el dispositivo Manos Libres Auriculares y acceder a sus funciones de audio:

- Capturar el audio recogido por el micrófono del dispositivo.



- Inyectar audio que sería reproducido por el auricular.



En algunos modelos vulnerables, incluso es posible cortar una conversación telefónica en curso e inyectar audio, para sorpresa del usuario.

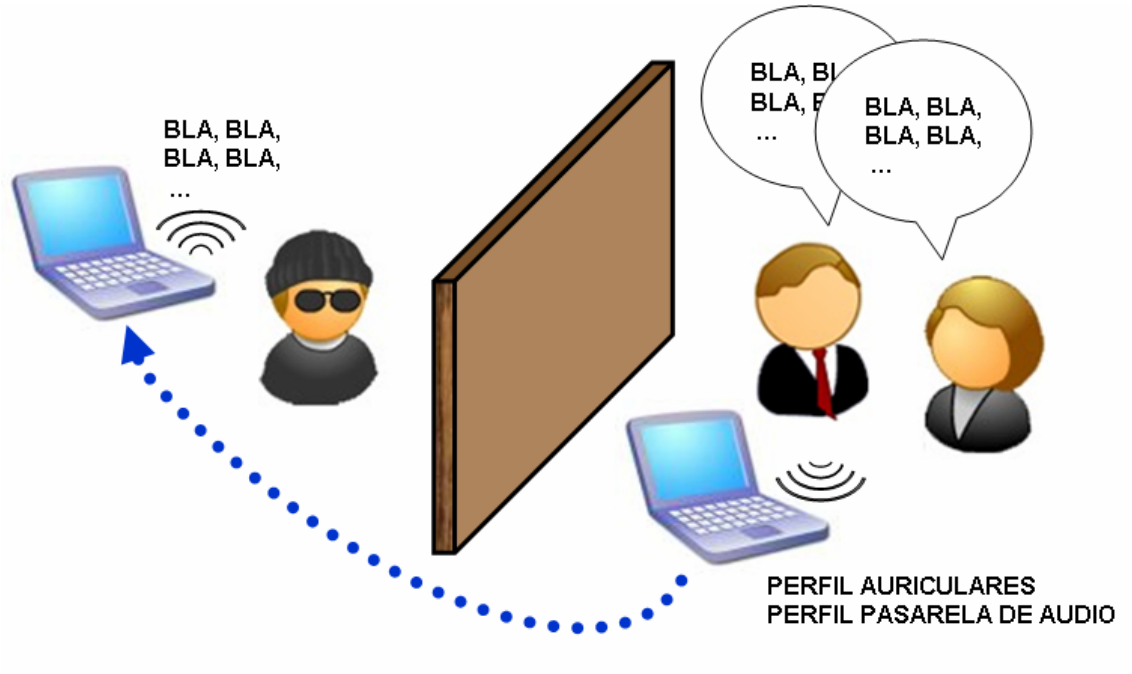
4.2.3 – The Laptop Whisperer (Kevin Finisterre, 2005)

El modelo de uso para el Perfil de Auriculares denominado *Pasarela de audio* permite a un usuario configurar dos equipos Bluetooth y establecer una pasarela de audio entre los dos, de forma que el audio que reproduce el software de un dispositivo, se transmite al otro dispositivo a través del enlace SCO y puede ser proyectado por los altavoces del segundo. Así mismo, el audio recogido por el micrófono de un dispositivo se transmite al otro dispositivo y puede ser grabado en un archivo de sonido.

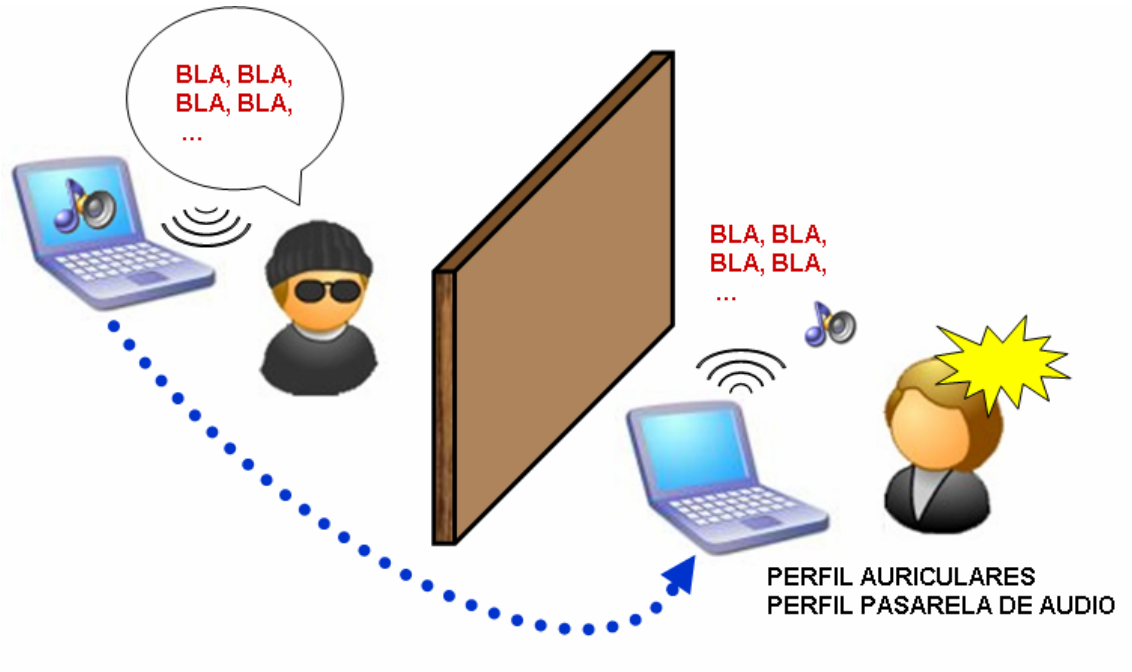
La esencia de la vulnerabilidad ***Laptop Whisperer*** se encuentra en la posibilidad de conectarse remotamente a un PC con Bluetooth y utilizar el Perfil de Auriculares sin necesidad de autenticación.

Un ataque con éxito al Perfil de Auriculares permitiría a un atacante llevar a cabo las siguientes acciones con fines maliciosos:

- Capturar el audio recogido por el micrófono del dispositivo, lo cual permitiría la escucha de conversaciones privadas.



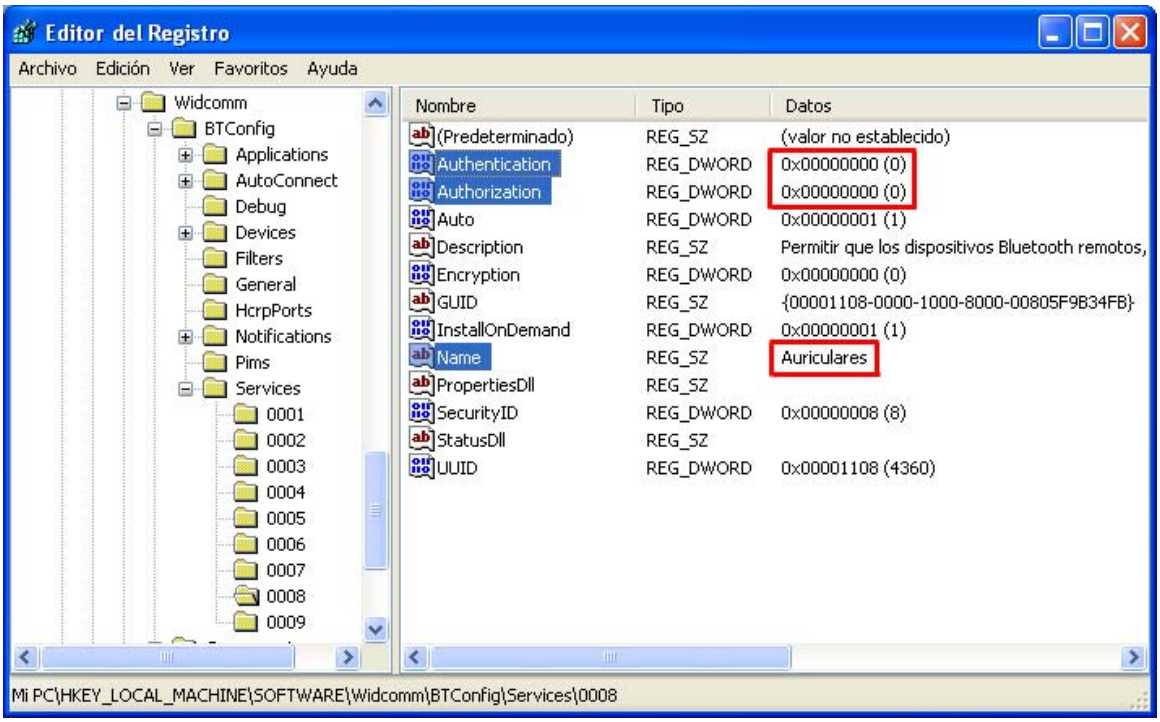
- Inyectar audio y mensajes de voz que serían reproducidos por los altavoces del PC, para sorpresa del usuario del mismo.



Las víctimas potenciales de un ataque ***Laptop Whisperer*** son todos aquellos usuarios de PC con Microsoft Windows™ que dispongan de un dispositivo Bluetooth soportado por la pila de protocolos Widcomm.

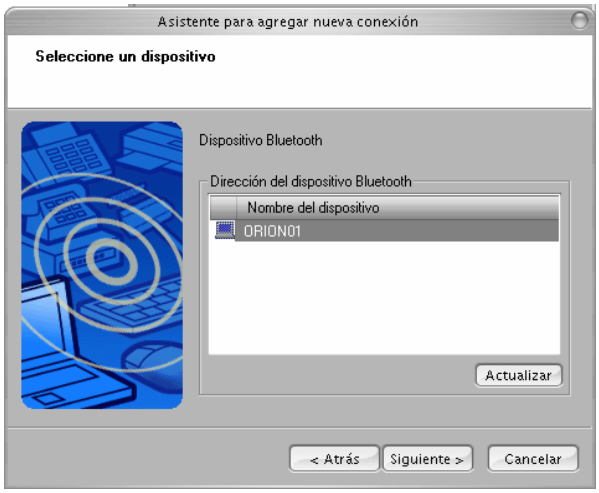
En el caso de usuarios con ordenador portátil o *laptop*, el caso es especialmente grave porque estos equipos suelen incorporar micrófonos que permitirían recoger audio de conversaciones próximas. Si un atacante consiguiera comprometer el Perfil de Auriculares, tendría acceso al control del micrófono y podría grabar conversaciones privadas. Si el equipo comprometido no incluyera micrófono, el atacante sólo podría limitarse a inyectar audio que sería proyectado por los altavoces del PC.

La vulnerabilidad que permite llevar a cabo el ataque ***Laptop Whisperer*** reside en la posibilidad de que un atacante remoto puede conectarse al Perfil de Auriculares sin necesidad de autenticación porque la configuración por defecto de la pila de protocolos Widcomm así lo establece.



El procedimiento para desarrollar el ataque ***Laptop Whisperer*** es el siguiente:

En primer lugar, el atacante debe localizar el PC objetivo y comprobar que soporta el Perfil de Auriculares.

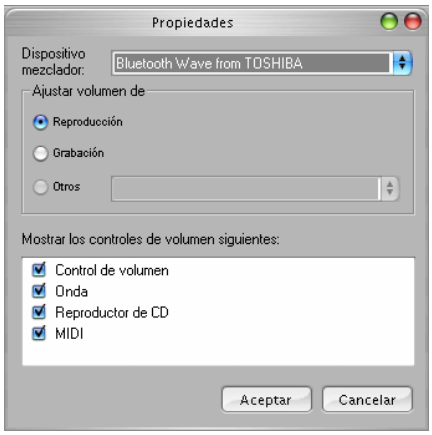


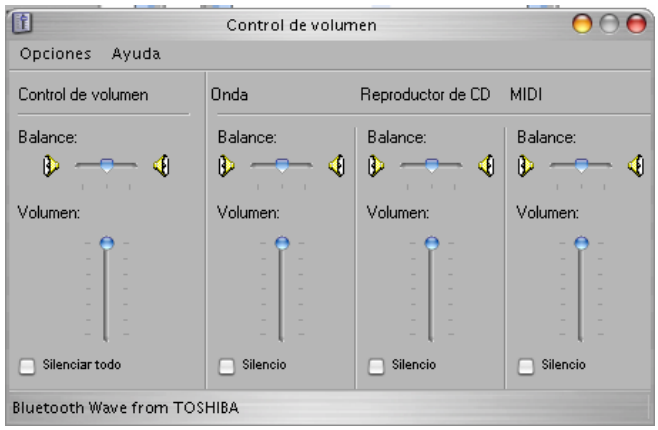
A continuación, el atacante establece una conexión con el Perfil de Auriculares del PC comprometido sin necesidad de autenticación.

La única notificación que recibe el usuario víctima es un aviso en el icono de Bluetooth, que cambia a color verde mientras existe una comunicación Bluetooth establecida con otro dispositivo. Esto significa que mientras el ataque se lleva a cabo, es muy improbable que el usuario perciba lo que realmente está ocurriendo.

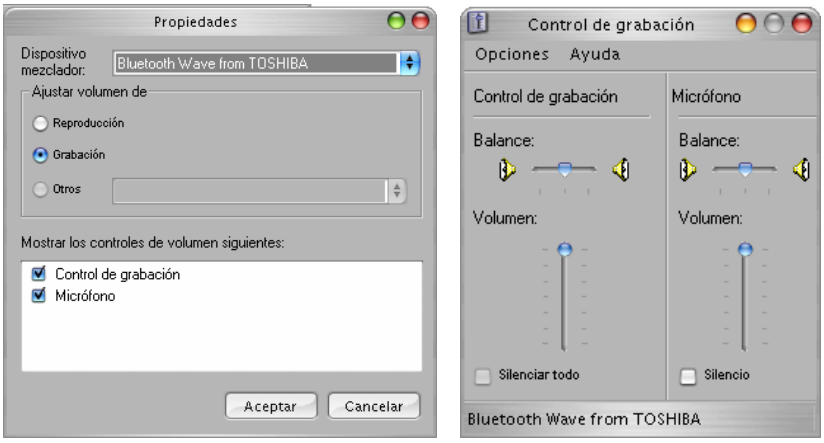
Con la conexión establecida, el atacante estará en disposición de:

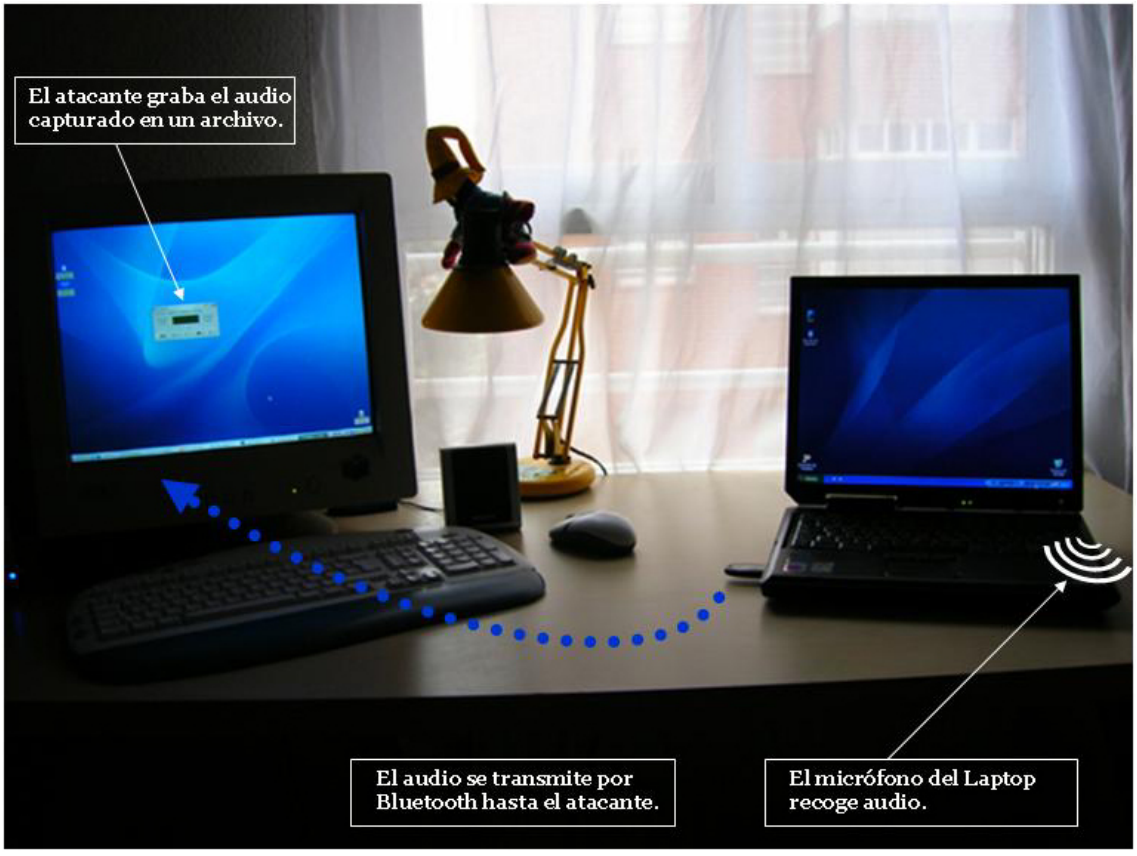
- Inyectar audio que será reproducido por los altavoces del PC comprometido.





- Capturar el audio recogido por el micrófono del Laptop y grabarlo en un fichero.





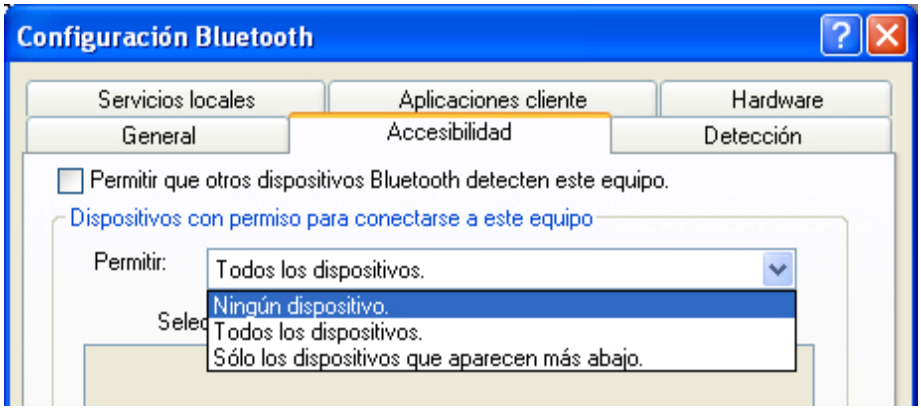
Para poder grabar el audio capturado desde el interfaz Bluetooth en un fichero se puede emplear la utilidad *sndrec32.exe* que incluye Windows™.

4.3 – Recomendaciones de seguridad para dispositivos Bluetooth

Se recomienda adoptar las siguientes medidas de seguridad con el fin de evitar ataques a dispositivos Bluetooth. Estas medidas son simples y de aplicación inmediata y deberían formar parte de la conducta habitual de un usuario de dispositivos Bluetooth.

1. Activar Bluetooth en el dispositivo sólo cuando sea necesario para realizar algún tipo de comunicación a través de este interfaz y desactivarlo cuando no se vaya a utilizar.

2. Configurar el dispositivo en modo oculto o *non discoverable*. De esta forma disminuyen las probabilidades de que un supuesto atacante detecte la presencia del dispositivo al escanear en búsqueda de equipos Bluetooth.

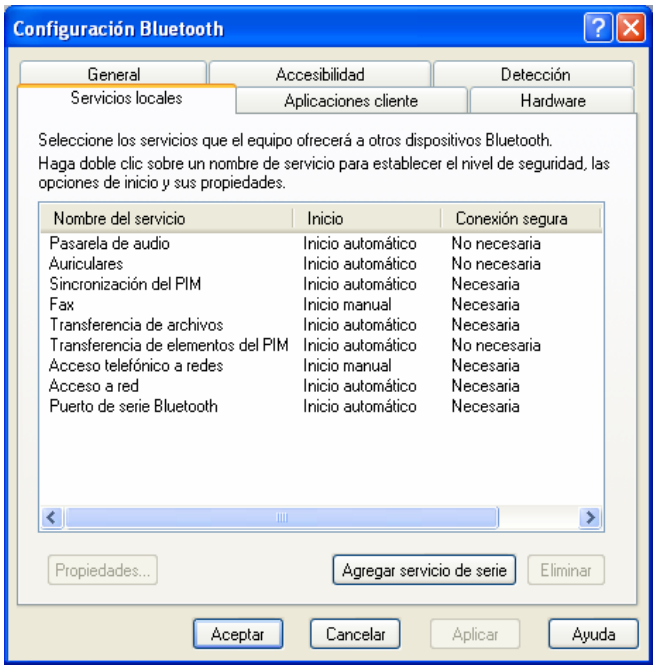


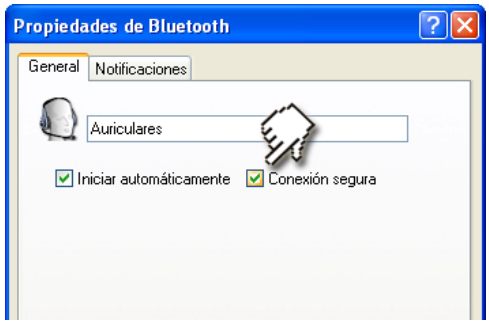
3. Configurar el dispositivo para que utilice la función de cifrado en todas las comunicaciones. De esto modo, se garantiza la confidencialidad del intercambio de mensajes.

4. Utilizar un nombre de dispositivo que no sea representativo de la marca y modelo del mismo, por ejemplo: *Nokia 6600.* Esto implica, en la mayoría de los casos, modificar el nombre de dispositivo asignado por el fabricante.

5. No aceptar bajo ningún concepto conexiones entrantes de dispositivos desconocidos. Esto implica también intentos de conexión de personas en las que no se confía aunque el pretexto pueda parecer inofensivo, por ejemplo: Emparejar dos dispositivos para transferir una fotografía.

6. Configurar todos los perfiles soportados por el dispositivo para que requieran autenticación ante cualquier intento de acceso. Esto es importante sobretodo para evitar ataques como ***The Laptop Whisperer,*** ya que, en algunos dispositivos, ciertos perfiles están configurados por defecto para admitir conexiones sin exigir autenticación.





7. Verificar periódicamente la lista de dispositivos de confianza y eliminar aquellas entradas de dispositivos con los que habitualmente no se establece conexión.

8. Aunque actualmente todavía no se ha descubierto una forma de romper la seguridad del emparejamiento realizando fuerza bruta sobre un código de seguridad Bluetooth (clave PIN) que hayan empleado dos dispositivos emparejados, utilizar en la medida de lo posible claves PIN de longitud extensa, hasta 16 bytes.

Capítulo

5

GO MOBILE!

5.1 – Ataques a dispositivos Bluetooth desde plataformas *mobile*

Existen algunas ventajas en la utilización de dispositivos *mobile*, tales como teléfonos móviles, PDAs y Tablet PCs, como plataformas de ejecución de herramientas de ataque a dispositivos Bluetooth. La ventaja principal de estos dispositivos, frente al uso de PCs, aunque se trate de ordenadores portátiles o *laptops*, es su capacidad de manejo.

Realizar un ataque de corto alcance a un dispositivo Bluetooth desde un ordenador portátil resulta incómodo y sospechoso. El uso de dispositivos *mobile* es más discreto y efectivo, nadie sospecharía de una persona que en ese momento está utilizando su teléfono móvil.

Esta ventaja se podría rebatir argumentando la posibilidad de llevar a cabo ataques automatizados desde un ordenador portátil guardado en una bolsa o mochila. Sin embargo, el hecho de poder manejar el dispositivo en cada momento permite al atacante disponer de mayor libertad de acción.

Otra ventaja que ofrecen los dispositivos *mobile* es que, actualmente, su capacidad de procesamiento es equiparable a la de cualquier PC. La mayoría de teléfonos móviles, PDAs y Tablet PCs del mercado disponen de procesadores y memoria suficientes para ejecutar una aplicación que permita llevar a cabo ataques contra dispositivos Bluetooth. Estas aplicaciones pueden ser desarrolladas en entornos embebidos gracias a SDKs (Software Development Kits) como ***eMbedded Visual C++*** o ***J2ME***.

Una posible desventaja es el radio de alcance de estos dispositivos *mobile*, los cuales sólo permiten alcanzar distancias cercanas a los 10 metros. Estas distancias, aún siendo significativamente inferiores a los 100 metros que pueden alcanzarse con un dispositivo USB de clase 1, permiten llevar a cabo ataques de corto alcance en espacios reducidos y con gran afluencia de personas, como bares, estadios, aeropuertos y lugares de interés turístico.

A continuación se describen algunos ejemplos de herramientas disponibles para dispositivos *mobile* y que permiten llevar a cabo ataques a dispositivos Bluetooth.

5.1.1 – Blooover (*Trifinite Group*, 2004)

Blooover es una herramienta prueba de concepto para realizar auditorías de seguridad en dispositivos Bluetooth desarrollada para teléfonos móviles con soporte J2ME.

Blooover permite llevar a cabo ataques ***Bluebug*** de forma automatizada sobre teléfonos móviles detectables mediante un simple escaneo de dispositivos Bluetooth. En caso de conseguir ejecutar un ataque con éxito, ***Blooover*** permite ejecutar las siguientes acciones:

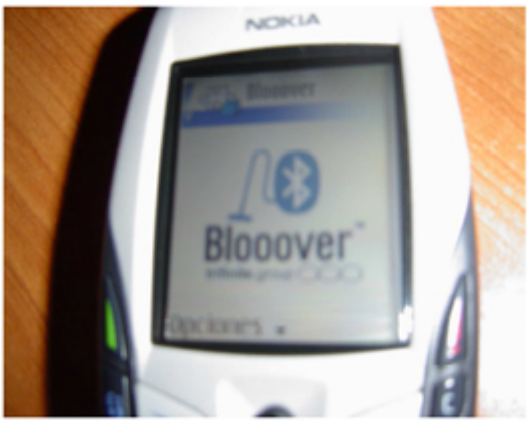
- *Snarf Phonebooks* - Captura de la agenda de contactos.
- *Snarf SMS* - Captura de mensajes SMS.
- *Add Phonebook Entry* - Añadir una entrada en la agenda de contactos.
- *Set Call Forward* - Establecer un desvío de llamadas a otro de teléfono.
- *Initiate Voice Call* - Iniciar una llamada de voz

Blooover está desarrollado en J2ME (JAVA 2 Micro Edition) y necesita que el teléfono móvil donde va a ser instalado sea compatible con MIDP 2.0 y el API Bluetooth JSR-82. Los siguientes modelos soportan los requerimientos necesarios para la ejecución de Blooover: Nokia™ 6600, Nokia™ 7610, Nokia™ 6630, Sony Ericsson™ P900, Siemens™ S65, etc.

La herramienta ***Blooover***, desarrollada por *Trifinite Group*, está disponible en la siguiente dirección: http://trifinite.org/trifinite_stuff_blooover.html

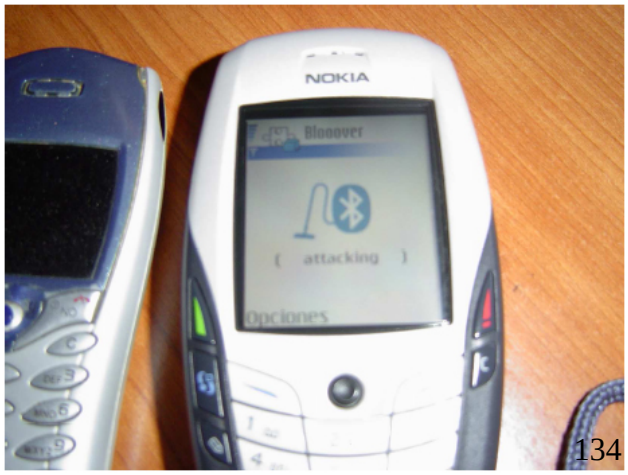
5.1.1.1 - Descripción del ataque Bluebug con Blooover

En primer lugar, activar Bluetooth en el teléfono móvil e iniciar ***Blooover***.



A continuación, el atacante puede visualizar y, en su caso, modificar las opciones de configuración que ofrece ***Blooover***

Finalmente, se inicia el ataque a través de la opción del menú principal *Find BT-Devices*. La aplicación comenzará a escanear en busca de dispositivos Bluetooth cercanos a los que lanzar el ataque.



Tras finalizar el ataque, ***Blooover*** genera un log con toda la información obtenida del teléfono móvil atacado.

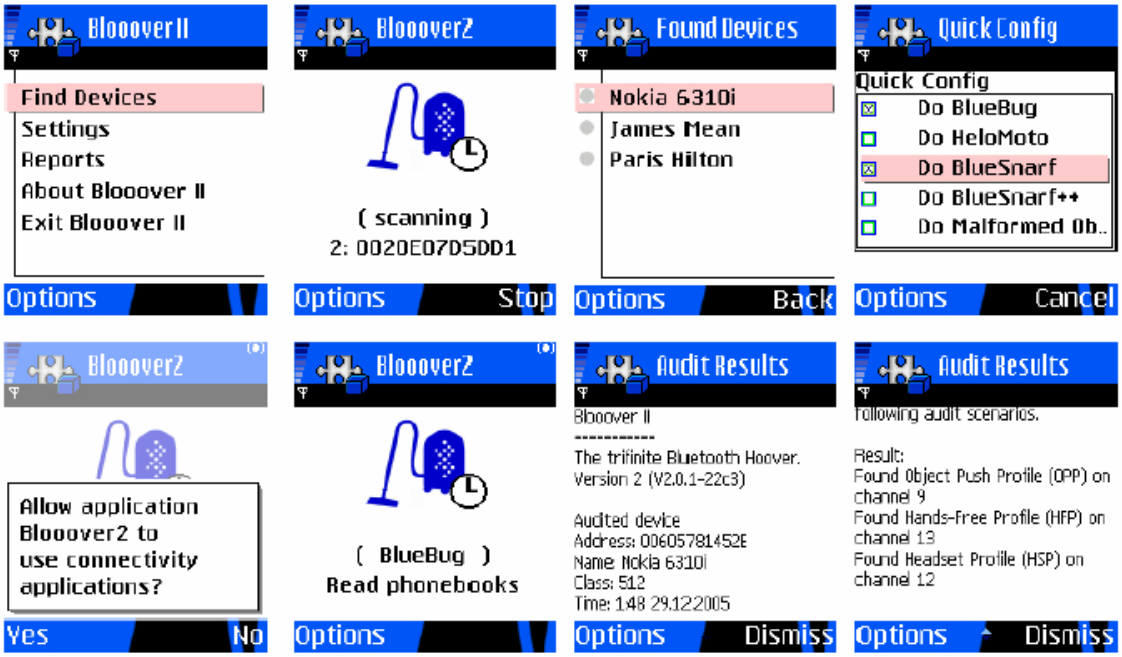
5.1.2 – Blooover II (*Trifinite Group*, 2005)

Blooover II es la siguiente generación de herramienta de auditoría de seguridad de teléfonos móviles desarrollada por *Trifinite Group*.

Incluye auditorías para las siguientes vulnerabilidades:

- ***Bluebug***
- ***HeloMoto***
- ***Bluesnarf***
- ***Bluesnarf++***
- Ataques de denegación de servicios (DoS, *Denial of Service*) mediante el envío de paquetes malformados por OBEX.

La herramienta ***Blooover II*** está disponible en la siguiente dirección:
http://trifinite.org/trifinite_stuff_bloooverii.html



Fuente: http://trifinite.org/Downloads/trifinite.presentation_22c3_berlin.pdf

Fuente: http://trifinite.org/Downloads/trifinite.presentation_22c3_berlin.pdf

5.1.3 – Pocket Bluesnarfer (Alberto Moreno, 2005)

Pocket Bluesnarfer es una herramienta para Pocket PC™ que permite conectarse al Perfil de Puerto Serie de un teléfono móvil y ejecutar comandos AT en el terminal con el fin de llevar a cabo las siguientes acciones:

- Ejecutar operaciones concretas mediante comandos AT personalizados.
- Obtener información general del teléfono móvil: marca, modelo, IMEI, cobertura y nivel de batería.
- Extraer la agenda de contactos almacenada en la tarjeta SIM.
- Extraer la bandeja de entrada de mensajes SMS.

El Perfil de Puerto Serie requiere autenticación en la mayoría de los teléfonos móviles, de modo que para poder ejecutar ***Pocket Bluesnarfer***, se requiere que el equipo Pocket PC™ y el teléfono móvil hayan sido emparejados previamente. Asimismo, ***Pocket Bluesnarfer*** se basa en conexiones RFCOMM emuladas sobre puertos COM virtuales, por lo que es posible que la aplicación no sea compatible con otras pilas de protocolos Bluetooth distintas de *Widcomm*.

Pocket Bluesnarfer está desarrollada en eMbedded Visual C++ con el SDK Windows Mobile™ 2002. Es compatible con Windows Mobile™ 2002 y 2003 y soporta las siguientes arquitecturas de Pocket PC™: ARM/Strong ARM/XScale

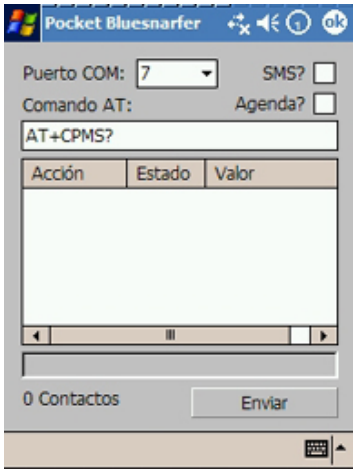
El código fuente y la herramienta ***Pocket Bluesnarfer*** están disponibles en la siguiente dirección: http://gospel.endorasoft.es/

Pocket Bluesnarfer está basado en el proyecto *IrDA Mobile*, de Daniel Strigl. http://www.codeproject.com/ce/irdamobile.asp

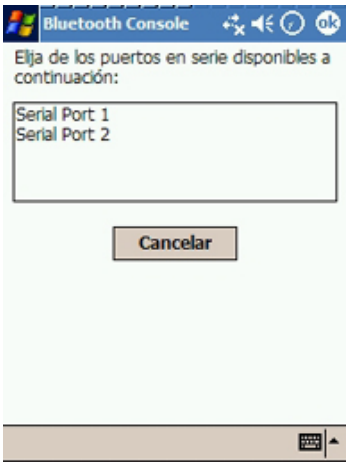
El código fuente se distribuye bajo licencia GNU General Public License.

5.1.3.1 - Descripción del ataque con *Pocket Bluesnarfer*

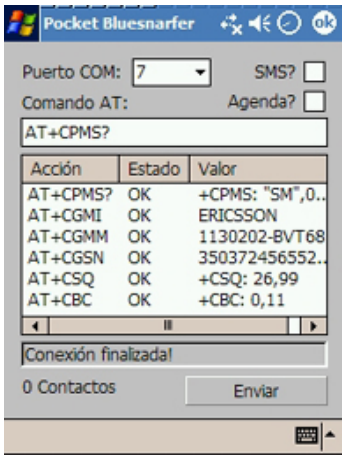
Antes de iniciar el ataque es necesario configurar el puerto COM que utilizará el equipo Pocket PC™ para comunicarse con el exterior (IrDA o Bluetooth). También se puede especificar el comando AT personalizado y elegir las opciones de extraer del teléfono móvil los mensajes SMS y la agenda de contactos.

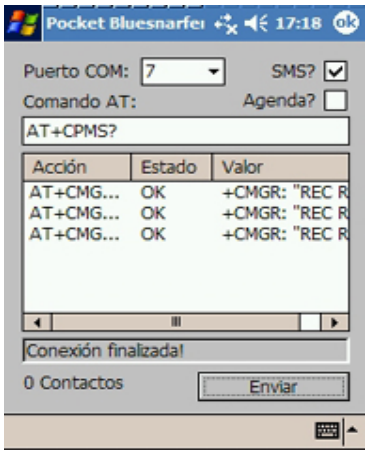


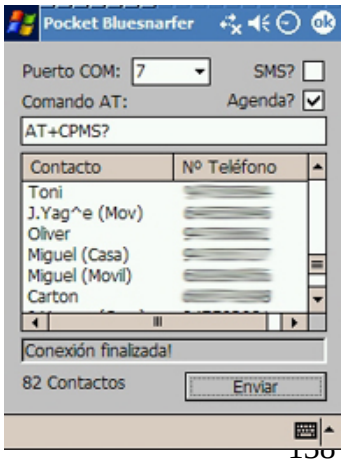
Al comenzar el ataque por Bluetooth, Windows Mobile™ 2003 lanzará el gestor de comunicaciones Bluetooth para localizar el dispositivo con el que se desea establecer comunicación. A continuación, el gestor de conexiones Bluetooth muestra los perfiles de Puerto Serie ofrecidos por el teléfono móvil.



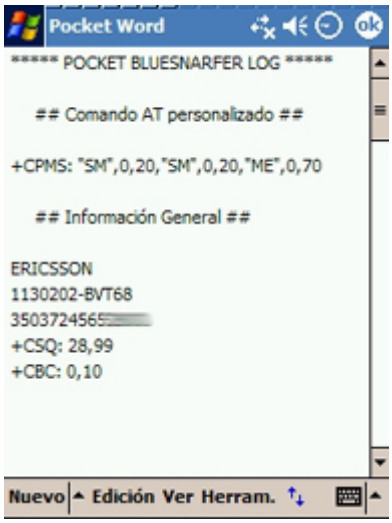
Al establecerse con éxito la conexión al Perfil de Puerto Serie, se inicia el ataque mediante comandos AT. El resultado se muestra en pantalla.

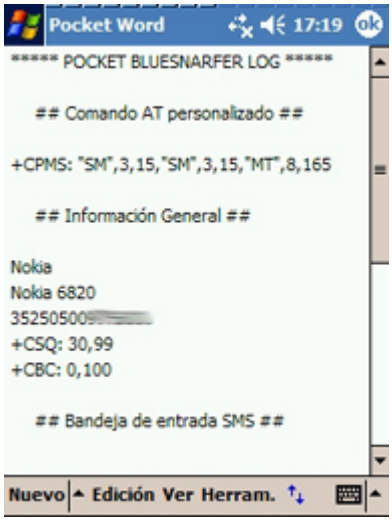


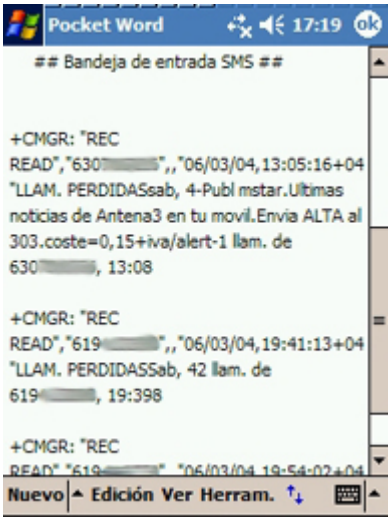




Finalizado el ataque, ***Pocket Bluesnarfer*** genera un archivo de log donde el usuario puede consultar la información extraída mediante la ejecución de comandos AT.





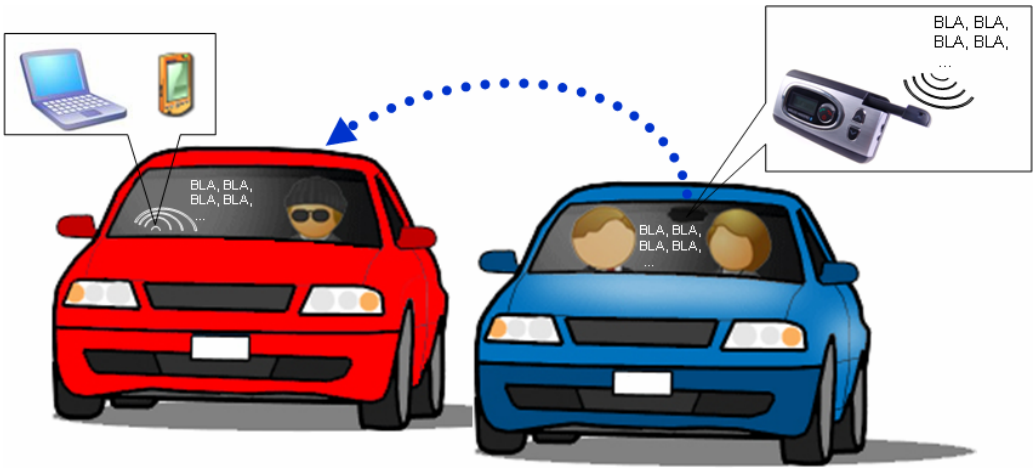


5.1.4 – The Pocket Car Whisperer (Alberto Moreno, 2005)

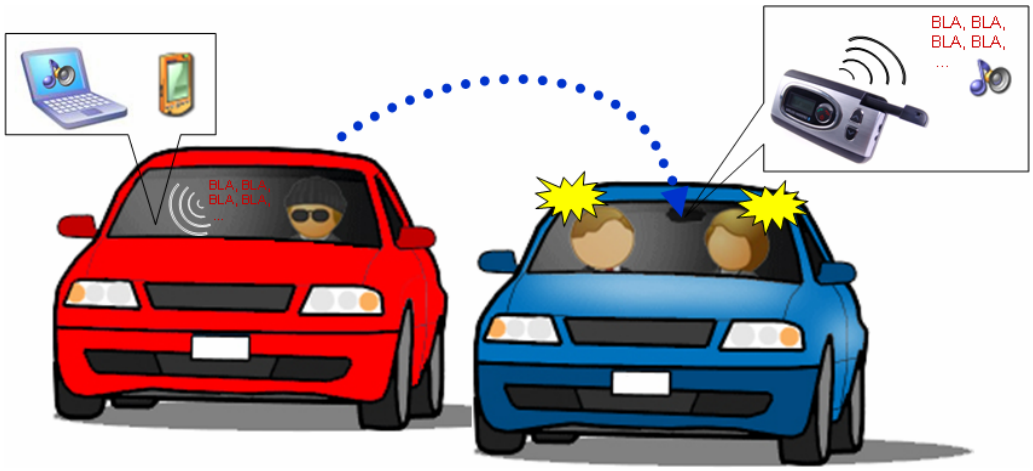
El ataque ***Car Whisperer***, desarrollado por *Trifinite Group*, explota la vulnerabilidad del código de seguridad Bluetooth (clave PIN) por defecto en dispositivos Manos Libres de automóvil. El ataque ***Pocket Car Whisperer*** es una implementación del ataque ***Car Whisperer*** para una plataforma atacante basada en Pocket PC™.

El hecho de incorporar una clave PIN por defecto en un dispositivo Bluetooth significa que cualquier usuario con conocimiento de esa clave estándar puede emparejarse con el dispositivo y comunicarse con él de forma autorizada. En el caso de un dispositivo Manos Libres, un atacante podría acceder a las funciones de audio implementadas en el terminal y llevar a cabo las siguientes acciones:

- Capturar el audio recogido por el micrófono del dispositivo, lo cual permitiría escuchar conversaciones privadas en el interior del vehículo.



- Inyectar audio que sería reproducido por los altavoces del dispositivo, lo cual permitiría proyectar mensajes de voz a los ocupantes del vehículo.



El ataque ***Car Whisperer*** original descubierto por el grupo ***Trifinite*** se basa en la utilización de una herramienta para Linux escrita en BlueZ, que se puede obtener en la dirección: http://trifinite.org/trifinite_stuff_carwhisperer.html

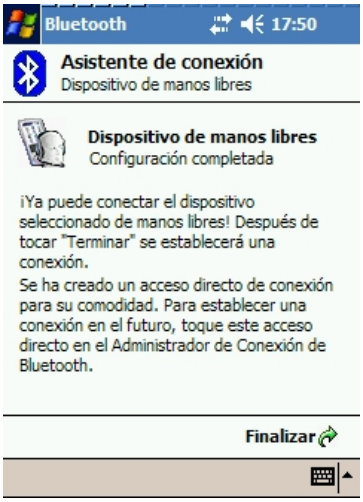
La variación del ataque ***Pocket Car Whisperer*** respecto del ataque original es el empleo de un equipo Pocket PC™ como plataforma de ataque. La ventaja principal, a parte de que una Pocket PC™ resulta más manejable y discreta que un ordenador portátil, es que el ataque puede ejecutarse sin necesidad de utilizar una herramienta diseñada específicamente para ese fin. Es posible llevar a cabo el mismo tipo de ataque mediante la combinación simple de varios elementos presentes en Windows Mobile™.

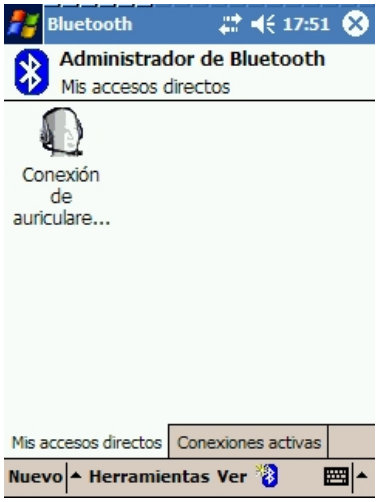
El procedimiento para desarrollar un ataque ***Pocket Car Whisperer*** es el siguiente:

En primer lugar, se procede a escanear en búsqueda de dispositivos Manos Libres Bluetooth.

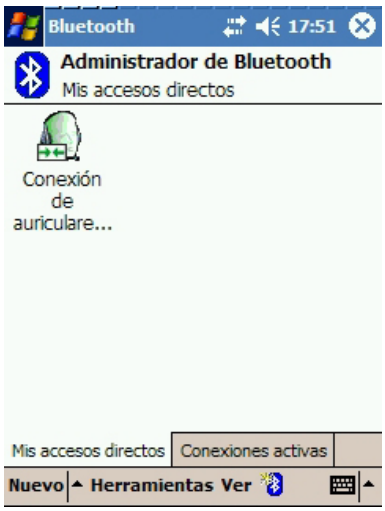
Tras encontrar un dispositivo Manos Libres, el siguiente paso es realizar el emparejamiento. Requiere conocer el código de seguridad Bluetooth (clave PIN) predeterminado que incluye el dispositivo por defecto: 1234, 0000, 8888…

Si el emparejamiento se ha producido satisfactoriamente, el gestor de conexiones Bluetooth crea un acceso directo al perfil de auriculares ofrecido por el dispositivo Manos Libres.





A continuación, el atacante debe establecer una conexión con el Perfil de Auriculares del dispositivo Manos Libres.



Si la conexión se ha realizado con éxito, el atacante dispondrá de acceso a las funciones de audio soportadas por el perfil de auriculares:

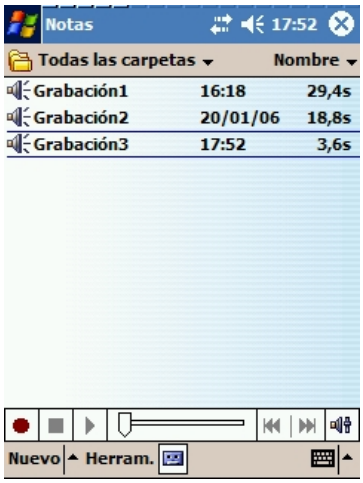
- Capturar el audio recogido por el micrófono del dispositivo.
- Inyectar audio que será reproducido por los altavoces del dispositivo.

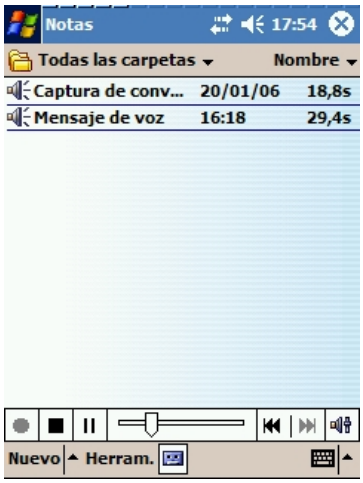
Para realizar estas funciones de audio, el atacante puede hacer uso de la aplicación *Micrófono* incluida en Windows Mobile™.
Una vez establecida la conexión al Perfil de Auriculares y creada una pasarela de audio estableciendo enlaces síncronos orientados a conexión (SCO) para la transmisión de audio en ambas direcciones, el atacante podrá llevar a cabo las siguientes acciones:

- Activar la aplicación Micrófono del equipo Pocket PC™ y capturar en un fichero el audio recogido por el micrófono del dispositivo Manos Libres. Esto permitiría tener acceso a conversaciones privadas entre los ocupantes del interior del vehículo objetivo.

- Grabar mensajes de voz por el micrófono del equipo Pocket PC™, que al ser reproducidos durante una sesión de ataque, se proyectarán en el interior del vehículo objetivo a través de los altavoces del dispositivo Manos Libres, con el consecuente sobresalto del conductor y del resto de pasajeros.

Adicionalmente, cualquier sonido o archivo de audio que se reproduzca en el equipo Pocket PC™ será proyectado a través de los altavoces del dispositivo Manos Libres. Esto significa que si el atacante reproduce un archivo de música en la aplicación Windows Media Player™, el sonido se escuchará en el interior del vehículo objetivo.





5.1.5 – Adaptación de herramientas de ataque a la plataforma Nokia™ 770 Internet Tablet (*Trifinite Group*)

El equipo especializado en seguridad en Bluetooth *Trifinite Group* descubrió y publicó las vulnerabilidades más importantes para teléfonos móviles Bluetooth: ***Bluebug***, ***Bluesnarf*** y ***HeloMoto***. Actualmente se encuentra en fase de desarrollo de aplicaciones de ataque a dispositivos Bluetooth para el equipo Tablet PC Nokia™ 770 Internet Tablet, basado en Linux.

Por ahora han portado a esta plataforma algunas herramientas de explotación de vulnerabilidades como ***Bluebug*** o ***HeloMoto*** utilizando el entorno de desarrollo *Maemo*; y tienen intención de seguir desarrollando más utilidades para este dispositivo *mobile* definitivo.

Fuente: http://trifinite.org/

Capítulo

6

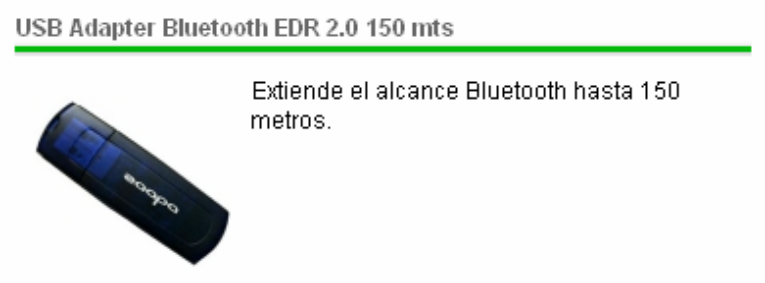
GO BEYOND!

6.1 – Aumento del alcance radio mediante antenas direccionales

El alcance radio de los módulos Bluetooth depende de su potencia de transmisión. Según la potencia de transmisión, los módulos Bluetooth se dividen en 3 clases:

- Clase 1: 100 mW / 20 dBm, con un rango de ~100 m.
- Clase 2: 2.5 mW /4 dBm, con un rango de ~10 m.
- Clase 3: 1 mW / 0 dBm, con un rango de ~1 m.

Los dispositivos USB Bluetooth convencionales suelen ser de clase 1 ó 2, con alcances de hasta 100 metros. No obstante, en la actualidad algunos fabricantes comercializan dispositivos USB Bluetooth con alcance de 125 y 150 metros.



Fuente: http://www.zaapa.co.uk

Sin embargo, 150 metros siguen siendo insuficientes para ataques de largo alcance, como ***The Car Whisperer***. El ataque ***Car Whisperer*** tiene como objetivo dispositivos Manos Libres de automóvil, de forma que resulta casi inviable para un atacante seguir a un objetivo en movimiento para llevar a cabo el ataque.

Por todo ello, han surgido técnicas de modificación de dispositivos Bluetooth que permiten realizar conexiones con antenas direccionales externas capaces de concentrar toda la emisión de la señal en una única dirección, ampliando considerablemente el alcance hasta distancias próximas a los kilómetros.

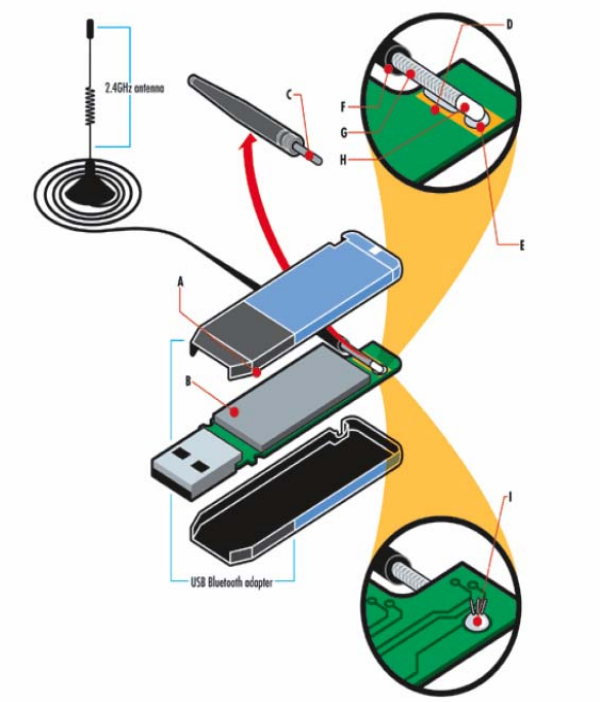
Algunos grupos dedicados al estudio de la seguridad en Bluetooth han publicado en la red manuales para poder llevar a cabo las modificaciones precisas en los módulos y procedimientos para la construcción de antenas direccionales capaces de amplificar al máximo el alcance de la señal Bluetooth.

La utilización de este tipo de dispositivos amplificadores de señal Bluetooth permitiría a los atacantes aumentar el número de dispositivos potenciales de recibir ataques como ***The Car Whisperer*** o ***The Laptop Whisperer***.

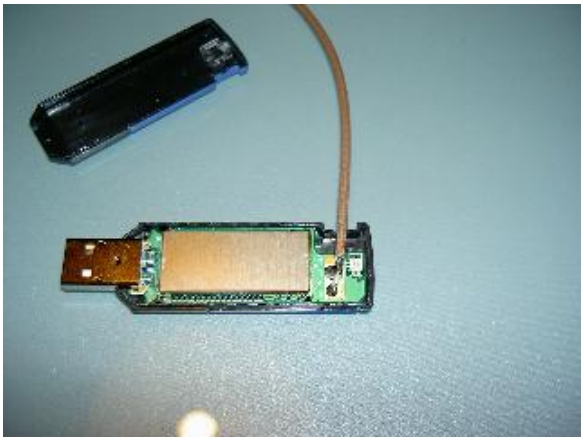
6.1.1 – Bluetooone (*Trifinite Group*, 2005)

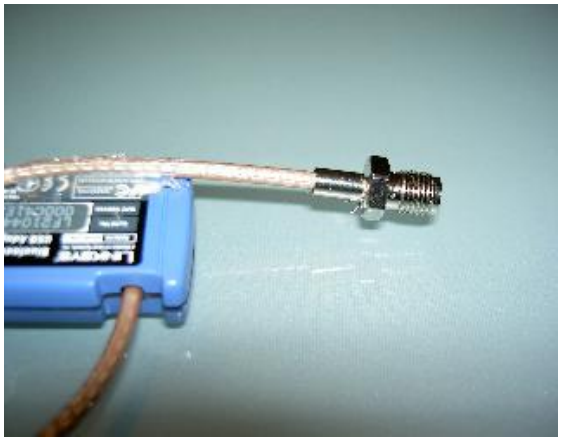
Bluetooone es un procedimiento desarrollado por *Trifinite Group* para conectar a un dispositivo USB Bluetooth una antena externa que permita aumentar su rango de cobertura y el número de objetivos potenciales a los que poder atacar.

En la página oficial http://trifinite.org/trifinite_stuff_bluetooone.html se proporcionan instrucciones detalladas para acoplar a un dispositivo Bluetooh un *pigtail* que permite conectarlo a una antena externa.



Fuente: http://trifinite.org/





Fuente: http://trifinite.org/

Gracias al procedimiento ***Bluetooone***, los miembros de *Trifinite Group* acoplan a sus dispositivos USB Bluetooth antenas direccionales muy potentes que les permiten desarrollar ataques como ***The Car Whisperer*** o ***Bluebug*** a dispositivos Bluetooth situados a gran distancia del atacante.

Por un lado, *Trifinite Group* ha conseguido desarrollar ataques ***Car Whisperer*** desde puntos estáticos situados en mitad de autopistas abarcando un gran número de vehículos potencialmente vulnerables.

Fuente: http://trifinite.org/

Por otro lado, un experimento llevado a cabo por *Trifinite Group* denominado ***Long Distance Snarf*** permitió efectuar un ataque ***Bluebug*** sobre un teléfono móvil situado a 1.78 kilómetros de distancia.

Fuente: http://trifinite.org/

6.1.2 – Bluesniper (Flexilis, 2004)

Bluesniper se define como una herramienta hardware de auditoría de seguridad en dispositivos Bluetooth. Consta de un rifle con mira telescópica que tiene acoplada una antena direccional capaz de emitir una señal Bluetooth hasta una distancia de 1.5 km.

El soporte de la antena en forma de rifle ha sido elegido por estética y comodidad en la sujeción, según los fabricantes, aunque no deja de ser un aparato llamativo.

Bluesniper fue presentado en la convención anual de hacking y seguridad informática *Defcon* 2004, que tuvo lugar en Las Vegas.

Fuente: http://www.flexilis.com/

Capítulo

7

FAKE HOT SPOTS:
EL CABALLO DE TROYA DE LA SEGURIDAD EN BLUETOOTH

El modelo de uso de Bluetooth *Marketing de proximidad por envío de publicidad* permite a algunas compañías llevar a cabo campañas publicitarias en las calles basadas en el envío masivo de publicidad directa al teléfono móvil a través de Bluetooth. Emplean dispositivos emisores colocados en puntos estratégicos de elevado tránsito de personas, capaces de enviar información personalizada que se adecua al modelo de teléfono móvil que recibe la información.

Aunque este modelo de uso Bluetooth fue inicialmente desarrollado por empresas con fines comerciales, algunos ayuntamientos han comprobado el éxito de este tipo de estrategias y han instalado sistemas de envío de información en puntos de interés general, como zonas turísticas, aeropuertos, intercambiadores de transporte público, edificios históricos y museos.

Fuente: http://www.futurlink.com/

Ya se han llevado a cabo con éxito proyectos pilotos en España, como la distribución del programa de eventos en la semana de fiestas de Bilbao, la posibilidad de descarga de la guía interactiva durante la feria SIMO TCI 2005 o la instalación en grandes ciudades de marquesinas con Bluetooth para el envío de *merchandising* relacionado con el estreno de una película.

Fuente: http://www.aecomo.org/

Debido al éxito de estas campañas publicitarias que emplean tecnología Bluetooth, su uso se está extendiendo en las grandes ciudades originando la aparición de una nueva moda entre los usuarios de teléfonos móviles: dejar Bluetooth activado en el dispositivo para recibir contenidos de publicidad mientras se transita por la ciudad.

Por supuesto, esta moda resulta peligrosa para los usuarios de teléfonos móviles, ya que exponen su dispositivo a potenciales ataques que afectan a vulnerabilidades en teléfonos móviles, como ***Bluebug***, ***Bluesnarf***, ***HeloMoto*** o ***Blueline***. A pesar de que la mayoría de teléfonos móviles modernos están protegidos frente a estas vulnerabilidades, algunos ataques siguen siendo factibles si el atacante logra conseguir la autorización para acceder a ciertos servicios que no requieren autenticación en el terminal.

En un escenario de zona urbana con abundantes campañas de marketing de proximidad basadas en Bluetooth, un atacante podría disfrazar su ataque bajo un envío de contenidos de publicidad, engañando al usuario del teléfono móvil para que autorizara la conexión con la esperanza de recibir algún paquete de información interesante. Si el atacante consiguiera obtener autorización para acceder a algún servicio no protegido, podría llegar a ser capaz de enviar virus y software malicioso a través del Perfil de Carga de Objetos (OBEX Object Push) o ejecutar comandos AT en el terminal como se describe en el ataque ***Blueline***.

La aparición en escena de gusanos para teléfonos móviles, como ***Cabir***, ***Commwarrior*** y ***Lasco***, que utilizan Bluetooth como medio de propagación, hace que sea factible desarrollar virus y gusanos para sistemas operativos de teléfonos móviles, como Symbian™ OS o Windows Mobile™, en forma de aplicaciones inofensivas tales como un salvapantallas o un tema para el interfaz del sistema.

En conclusión, el uso de campañas de marketing de proximidad como medio para disfrazar envíos de objetos maliciosos a teléfonos móviles hace que puedan diseñarse aplicaciones denominadas *troyanos* que ejecutan código maligno bajo la apariencia de un contenido de publicidad atractivo.

La recomendación que ofrece el autor de este Proyecto Fin de Carrera es evitar caer en la moda de dejar Bluetooth activado en el teléfono móvil por defecto y sólo activarlo en caso de necesidad para llevar a cabo una comunicación con otro dispositivo de confianza. Así mismo, se recomienda no aceptar conexiones provenientes de dispositivos no conocidos aunque éstas se presenten como contenidos de publicidad inofensivos.

Capítulo

8

VALORACIÓN ECONÓMICA

Este proyecto de "Seguridad en Bluetooth" es un estudio general de los aspectos relacionados con la seguridad del estándar de comunicaciones Bluetooth y su implementación en dispositivos. Por tratarse de un estudio científico, no tiene sentido comercializar la idea y los únicos productos que se han obtenido durante el desarrollo de este proyecto son programas *open-source* distribuídos bajo licencia GNU.

Sin embargo, para la realización del proyecto se ha necesitado disponer de una gran variedad de equipos informáticos y dispositivos electrónicos, que se detallan a continuación:

PC de sobremesa (~ 1400 €)
PC portátil (~ 1200 €)
Pocket PC™ Dell Axim x30 (~ 300 €)
Teléfono móvil Nokia™ 6600 (~ 150 €)
Teléfono móvil Sony-Ericsson™ T68 (~ 60 €)
Dispositivo Manos Libres Bluetooth MiniTooth (~ 84 €)
Dispositivo USB Bluetooth IOGEAR™ (~ 25 €)
Dispositivo USB Bluetooth Zaapa™ (~ 30 €)

En caso de querer llevar a cabo un estudio de las mismas caracterísiticas, sería necesario disponer de un equipamiento informático y electrónico similar.

Todas las herramientas de auditoría de seguridad empleadas en el proyecto se distribuyen libremente y el desarrollo software se ha llevado a cabo en Linux Fedora Core 5.

Capítulo

9

CONCLUSIONES DEL PROYECTO

El estudio realizado por el proyecto "Seguridad en Bluetooth" ha demostrado que los primeros modelos de dispositivos Bluetooth carecían de mecanismos robustos de seguridad y presentaban vulnerabilidades que permitían a un atacante comprometer el dispositivo, afectando a la privacidad del usuario propietario y a la integridad del equipo.

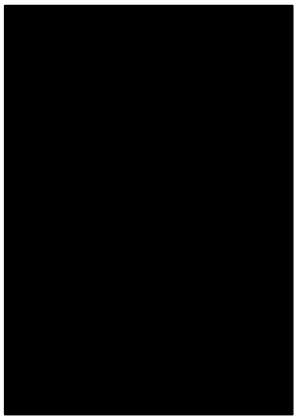
Se ha demostrado que estas vulnerabilidades se deben principalmente a la escasa preocupación de los fabricantes por implantar mecanismos de seguridad en los dispositivos que comercializan y se destaca el hecho de que el protocolo Bluetooth es uno de los más robustos que existen, ya que ofrece mecanismos de autenticación y cifrado de datos que aseguran el intercambio de comunicaciones entre dispositivos.

A lo largo del proyecto se han documentado vulnerabilidades en dispositivos tales como teléfonos móviles y equipos Manos Libres así como técnicas de ataque desarrolladas por equipos dedicados al estudio de la seguridad en Bluetooth. Estas técnicas de ataque se basan en la utilización de herramientas capaces de automatizar el proceso de explotación de una vulnerabilidad y que permiten al atacante obtener información sensible y comprometer la privacidad de usuarios de dispositivos.

Con el fin de evitar ser víctima de un ataque por medio de dispositivos Bluetooth, se han dictado unas recomendaciones para el buen uso de la tecnología Bluetooth en aquellos equipos que la incorporan. Se trata de normas simples y de aplicación inmediata que deberían formar parte de la conducta habitual de un usuario de dispositivos Bluetooth.

Por último, se ha querido hacer especial hincapié en el uso de Linux como plataforma avanzada para el establecimiento de comunicaciones Bluetooth, ya que ofrece una pila de protocolos, BlueZ, sencilla de manejar, con multitud de herramientas a disposición del usuario y con un entorno de desarrollo que simplifica la programación de potentes aplicaciones.

Como líneas de investigación futuras, se propone continuar con el estudio de vulnerabilidades en los nuevos productos Bluetooth comercializados y la publicación de técnicas de ataque que permitan explotar esas fallas de seguridad como único medio para que tanto los fabricantes como los usuarios tomen conciencia del riesgo que supone la comercialización de dispositivos Bluetooth vulnerables.



BIBLIOGRAFÍA Y REFERENCIAS

Bibliografía

[MULL02] Muller, Nathan J., "Tecnología Bluetooth", McGraw-Hill 2002.

Referencias On Line

www.bluetooth.com – The Official Bluetooth Web site

www.bluetooth.org – The Official Bluetooth Membership Site

www.palowireless.com/bluetooth/ - Bluetooth Resource Center

www.bluez.org - Official Linux Bluetooth protocol stack

en.wikipedia.org/wiki/Bluetooth – Wikipedia, the free enciclopedia

es.wikipedia.org/wiki/Bluetooth – Wikipedia, la enciclopedia libre

developer.apple.com – Apple Developer Connection

discussion.forum.nokia.com – Nokia Developer Discusión Boards

people.csail.mit.edu/albert/bluez-intro/ - Bluetooth programming in GNU/Linux

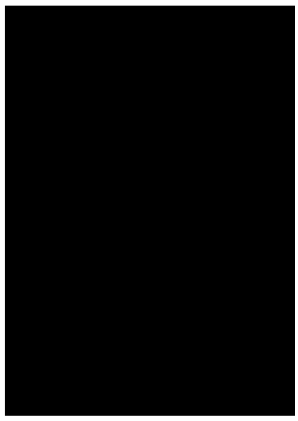
trifinite.org – the home of the trifinite.group

www.thebunker.net/security/bluetooth.htm - The Bunker | Security | Bluetooth

www.digitalmunition.com – Digital Munition

bluehack.elhacker.net – Bluehack: the Spanish Bluetooth Security Group

gospel.endorasoft.es – El Blog de Gospel - Seguridad en Bluetooth



ANEXOS

Anexo I – Juego de comando AT GSM

Notación empleada en las definiciones:

Comando AT: [Definición técnica]
- Funcionalidad del comando
- Sintaxis: Petición | Respuesta
- Respuesta obtenida al comando

Comandos AT para operaciones básicas

AT+CPAS: [Phone Activity Status]
1) AT+CPAS=?
- Muestra la implementación del comando.
- Sintaxis: AT+CPAS=? | +CPAS: (lista de estados soportados)
0 - Ready (Encendido pero inactivo)
1 - Unavailable (No disponible)
2 - Unknown (Desconocido)
3 - Ringing (Llamada entrante en proceso)
4 - Call in progress (Llamada saliente en proceso)
5 - Asleep (Dormido)
- Respuesta: +CMGD: (0,2,3,4)

2) AT+CPAS
- Informa del estado de actividad del teléfono.
- Sintaxis: AT+CPAS | +CPAS: <estado>
- Respuesta: +CPAS: 0, en estado normal de inactividad.
+CPAS: 3, si el teléfono atacado está sonando a causa de una llamada entrante.

ATD: [Dial Command]
- Inicia una llamada telefónica
- Sintaxis: ATD64612345 para una llamada de Datos.
ATD64612345; para una llamada de Voz. (Importante ;)
ATD>"Alberto"; para llamar al contacto almacenado en la agenda con el texto asociado Alberto.

AT+CCFC: [Call Forwarding Number]
- Gestiona el Desvío de Llamadas. Permite redireccionar llamadas entrantes a otro número de teléfono.
- Sintaxis:
AT+CCFC=<razón>,<modo>,<número>,<tipo>,<clase>,[<subaddr>,<satype>,[<time>]]

<razón> Razón por la cual entra en acción el desvío de llamada.
0 - incondicional
1 - si teléfono ocupado
2 - si no obtiene respuesta
3 - si inalcanzable
4 - todos los desvíos de llamadas
5 - todos los desvíos de llamadas condicionales

<modo> Estado del desvío de llamada.
0 - desahabilitado
1 - habilitado
2 - query status
3 - registro
4 - erasure (borrado)

<número> Cadena de texto con el número de teléfono destino del desvío de llamada. Se especifica en el formato indicado en el campo <type>

<tipo> Tipo de código de dirección de teléfono:
145 - para código internacional +
129 - en otro caso

<clase> Código que representa la clase de información que contiene la llamada a desviar.
1 - voz
2 - datos
4 - fax
7 - cualquier clase (por defecto)

<time> Tiempo en segundos a esperar antes de desviar la llamada.
1..30 (por defecto, 20)

<status> Estado de la opción desvío de llamadas. (Sólo en respuesta AT)
0 - no activo
1 – activo

- Ejemplo: Implementación del comando en Blooover:
"AT+CCFC=0,3,\"+4913377001\",145,7\r"
Vemos que utiliza los siguientes parámetros:
<razón> = 0, incondicional
<modo> = 3, registro
<número> = +4913377001
<tipo> = 145, formato de código internacional
<clase> = 7, cualquier clase de información a desviar

AT+CGMI: [Request Manufacturer Identification]
- Petición de identificación del Fabricante (Marca del teléfono).
- Sintaxis: AT+CGMM | <fabricante>
- Respuesta: Nokia Mobile Phones

AT+CGMM: [Request Model Identification]
- Petición de identificación del modelo de teléfono.
- Sintaxis: AT+CGMM | <modelo>
- Respuesta: Nokia 6820

AT+CGSN: [Request Product Serial Number Identification]
- Petición de identicación del número de serie del producto.
- Sintaxis: AT+CGSN | <IMEI>
- Respuesta: 1234567890987654321 (IMEI)

AT+CBC: [Battery Charge]
- Devuelve el estado de carga de la batería.
- Sintaxis: AT+CBC | +CBC: <bcs>, <bcl>
<bcs> = 0 indica que el teléfono está conectado a una batería.
<bcl> = 0 indica que el teléfono tiene la batería agotada.
= 1..100 indica el porcentaje de carga que aún queda por agotar.
- Respuesta: +CBC:0,56

AT+CSQ: [Signal Quality]
- Devuelve el estado de calidad de la señal de cobertura.
- Sintaxis: AT+CSQ | +CSQ: <rssi>,<ber>
<rssi> = 0 indica -113 dBm o menos
= 1 indica -111 dBm
= 2..30 indica -109..-53 dBm
= 31 indica -51dBm o más
= 99 indica desconocido
<ber> = 99 indica porcentaje desconocido
- Respuesta: +CSQ: 13,9

Comandos AT para gestión de la agenda de contactos

AT+CPBS: [Select Phone Book Memory Storage]
1) AT+CPBS?
- Informa de los dispositivos de memoria que soporta el teléfono para almacenar las distintas listas de contactos.
- Sintaxis: AT+CPBS? | +CPBS: "XX", donde "XX" se sustituye por el dispositivo de almacenamiento:
"SM" - SIM phonebook list [Lista de contactos de la agenda SIM]
"TA" - TA phonebook list [Lista de contactos del terminal]
"LD" - SIM last dialing list [Lista de números marcados]
"DC" - Dialled call list [Lista de llamadas realizadas]
"RC" - ME received calls list [Lista de llamadas recibidas]
"MC" - ME missed call list [Lista de llamadas perdidas]
"EN" - Emergency number list [Lista de números de emergencia]
"FD" - SIM fix dialing list
"MT" - ME + SIM combined list
"ON" - SIM o ME own number list
- Respuesta: +CPBS: "SM"

2) AT+CPBS="XX"
- Selecciona por defecto uno de los dispositivos de memoria que soporta el teléfono para almacenar las listas de contactos.
- Sintaxis: AT+CPBS="XX", donde "XX" se sustituye por el dispositivo de almacenamiento:
"SM" - SIM phonebook list [Lista de contactos de la agenda SIM]
"TA" - TA phonebook list [Lista de contactos del terminal]
"LD" - SIM last dialing list [Lista de números marcados]
"DC" - Dialled call list [Lista de llamadas realizadas]
"RC" - ME received calls list [Lista de llamadas recibidas]
"MC" - ME missed call list [Lista de llamadas perdidas]
"MT" - ME + SIM combined list
"ON" - SIM o ME own number list

AT+CPBR: [Read Phone Book Entry]
1) AT+CPBR=?
- Informa del tamaño de la agenda de contactos.
- Sintaxis: AT+CPBR=? | +CPBR: <(1-n)>,<nlen>,<tlen>
<(1-n)> indica el rango de índices que la agenda puede contener.
<nlen> indica la longitud máxima permitida para un número de teléfono.
<tlen> indica la longitud máxima permitida para el texto asociado a ese número (nombre del contacto).
- Respuesta: +CPBR: (1-150),48,14

2) AT+CPBR=<indice>
- Leer una entrada de la agenda de contactos.
- Sintaxis: AT+CPBR=<indice inicial> [,<indice final>] | +CPBR: <índice>, <número>, <tipo>, <texto>
<índice> indica el índice de la agenda de contactos.
<número> indica el número de teléfono almacenado en el índice.
<tipo> indica el tipo de número de teléfono 129 o 145 si incluye el prefijo internacional +.
<text> indica el texto asociado al número de teléfono, normalmente, el nombre del contacto.
- Respuesta a AT+CPBR=8: +CPBR: 8,"646123456",129,"Pepito"

AT+CPBS;+CPBR [Leer una entrada de una lista de contactos seleccionada]
- Primero elegimos la lista de contactos a la que queremos acceder, y luego leemos una entrada por su índice.
Sintaxis: AT+CPBS="XX";+CPBR=<índice>, donde "XX" se sustituye por el dispositivo de almacenamiento:
"SM" - SIM phonebook list [Lista de contactos de la agenda SIM]
"TA" - TA phonebook list [Lista de contactos del terminal]
"LD" - SIM last dialing list [Lista de números marcados]
"DC" - Dialled call list [Lista de llamadas realizadas]
"RC" - ME received calls list [Lista de llamadas recibidas]
"MC" - ME missed call list [Lista de llamadas perdidas]
"EN" - Emergency number list [Lista de números de emergencia]
"FD" - SIM fix dialing list
"MT" - ME + SIM combined list
"ON" - SIM o ME own number list
- Ejemplo de Respuesta a AT+CPBS="DC";+CPBR=2: +CPBR: 2,"646123456",129,"Alberto"
(Visualizamos el último contacto al que hemos llamado).
AT+CPBS="MC";+CPBR=1: +CPBR: 1,"646987654",129,"Alberto"
(Visualizamos la última llamada perdida).

AT+CPBF: [Find Phone Book Entries]
- Devuelve la entrada de la agenda de contactos cuyo texto asociado a un número contiene la cadena alfanumérica proporcionada.
- Sintaxis: AT+CPBF="textoaencontrar" | +CPBR: <índice>, <número>, <tipo>, <texto>
"textoaencontrar" es case-sensitive
<índice> indica el índice de la agenda de contactos.
<número> indica el número de teléfono almacenado en el índice.
<tipo> indica el tipo de tipo de número de teléfono. Por defecto, 129 o 145 si incluye el prefijo internacional +.
<text> indica el texto asociado al número de teléfono, normalmente, el nombre del contacto.
- Respuesta a AT+CPBF="Alberto": +CPBF: 19, "646987654",129,"Alberto"

AT+CPBW: [Write Phone Book Entry]
- Escribe una entrada en la agenda de contactos.
- Sintaxis: AT+CPBW = <índice>, <número>, <tipo>, <texto>
<índice> indica el índice de la agenda de contactos donde se creará la entrada de contacto. Si no se proporciona índice, se añade la entrada en el primer hueco libre.
<número> indica el número de teléfono almacenado en el índice.
<tipo> indica el tipo de tipo de número de teléfono. Por defecto, 129 o 145 si incluye el prefijo internacional +.
<text> indica el texto asociado al número de teléfono, normalmente, el nombre del contacto.
Nota: Si únicamente se proporciona el campo del índice (omitiendo el resto de campos), la entrada de la agenda asociada a ese índice se borrará.
- Ejemplo para crear un nuevo contacto:
AT+CPBW=,"696224466",129,"Alberto"

Comandos AT para gestión de mensajes SMSs

AT+CMGF: [Message Format]
1) AT+CMGF=?
- Informa de los formatos de mensaje soportados por el teléfono
- Sintaxis: AT+CMGF=? | +CMGF: (0,1)
modo = 0 indica formato de mensajes en modo PDU
modo = 1 indica formato de mensajes en modo TEXTO

2) AT+CMGF?
- Informa del formato de mensajes que está siendo utilizado actualmente para los comandos enviar, listar, leer y escribir.
- Sintaxis: AT+CMGF? | +CMGF: <modo>
modo = 0 indica formato de mensajes en modo PDU
modo = 1 indica formato de mensajes en modo TEXTO

3) AT+CMGF=<modo>
- Establace el formato a utilizar para la entrada y salida de mensajes.
- Sintaxis: AT+CMGF=<modo>
modo = 0 indica formato de mensajes en modo PDU
modo = 1 indica formato de mensajes en modo TEXTO

AT+CMGL: [List Messages]
1) AT+CMGL=?
- Informa de los posibles estados de un mensaje en la memoria que el teléfono puede soportar.
- Sintaxis: (+CMGF=0) AT+CMGL=? | +CMGL: (0-4)
(+CMGF=1) AT+CMGL=? | +CMGL: ("REC UNREAD", "REC READ", "STO UNSENT", "STO SENT", "ALL")
<estados>:
0 | "REC UNREAD": Almacenado en Bandeja de entrada y sin leer.
1 | "REC READ": Almacenado en Bandeja de entrada y leído.
2 | "STO UNSENT": Almacenado en Bandeja de salida y sin enviar.
3 | "STO SENT": Almacenado en Bandeja de salida y enviado.
4 | "ALL": Todos los mensajes almacenados.

2) AT+CMGL=<estado>
- Lista todos los mensajes almacenados correspondientes al estado especificado.
- Sintaxis: AT+CMGL=<estado> | +CGML: <índice>, <estado>, <número>, [otros parámetros opcionales] ,
<timestamp><CR><LF><Cuerpo del mensaje>
<indice> Posición que ocupa el mensaje SMS en la memoria.
<estado> Estado del mensaje.
<número> Cadena de texto con el número de teléfono origen del mensaje.
<timestamp> Fecha y hora.
<CR><LF> Retorno de carro y salto de linea.
- Resultado a AT+CMGL="STO UNSENT":
+CMGL: 16,"STO UNSENT","679123456",,
Hola! Esto es un sms almacenado en memoria. Luego puede ser enviado... Salu2

AT+CMGR: [Read Message]
- Permite leer mensajes SMS de la bandeja de entrada.
- Sintaxis: AT+CMGR=<índice> | +CMGR: <estado>, <número>, [otros parámetros] , <timestamp><CR><LF><Cuerpo del mensaje>
<indice> Posición que ocupa el mensaje SMS en la memoria.
<estado> Estado del mensaje.
<número> Cadena de texto con el número de teléfono origen del mensaje.
<timestamp> Fecha y hora.
<CR><LF> Retorno de carro y salto de linea
- Respuesta a AT+CMGR=1:
+CMGR: "REC READ","227",,"05/07/12,14:13:02+08"
Movistar info: Ahora, GRATIS, tus Llamadas Perdidas vienen con el NOMBRE de la persona que te llamo, si esta en tu agenda. Para volver al SMS del 200 llama 283

AT+CMGD: [Delete Message]
1) AT+CMGD=?
- Muestra la implementación del comando.
- Sintaxis: AT+CMGD=? | +CMGD: (lista de índices soportados)[,(lista de delflags soportadas)]
- Respuesta: +CMGD: (1-15),(0-4) //1-15 indica que la memoria SIM puede almacenar de 1 a 15 mensajes SMS

2) AT+CMGD=<índice>
- Elimina el mensaje de índice especificado.
- Sintaxis: AT+CMGD=1 elimina el mensaje con índice 1, es decir, el primer mensaje de la bandeja de entrada de mensajes SMS.

AT+CMGW: [Write Message]
- Permite escribir un mensaje SMS en memoria (Bandeja de salida).
- Sintaxis: AT+CMGW=<número> [Presionar CR]
> Escribimos el cuerpo del mensaje. [Presionar Ctrl-Z]

<número> Cadena de texto con el número de teléfono destino del mensaje.
- Ejemplo:
AT+CMGW="679123456" [Presionar CR]
> Hola! Como estás? Hace mucho q no te veo. Ciao. [Ctrl-Z]
Respuesta: +CMGW: <índice que ocupará en la memoria>

AT+CMGS: [Send Message]
- Permite enviar un mensaje SMS.
- Sintaxis: AT+CMGS=<número> [Presionar CR]
> Escribimos el cuerpo del mensaje. [Presionar Ctrl-Z]

<número> Cadena de texto con el número de teléfono destino del mensaje.
- Ejemplo:
AT+CMGS="679123456" [Presionar CR]
> Hola! Como estás? Hace mucho q no te veo. Ciao. [Ctrl-Z]
Respuesta: +CMGS: 213