

SEGURIDAD EN BLUETOOTH

Alejandro Ramos, CISSP

SIA

aramosf@sia.es

RESUMEN

Bluetooth se ha visto fuertemente impulsado debido a la integración con la telefonía móvil y sus múltiples aplicaciones. Como siempre sucede cuando se populariza un protocolo, comienza a ser estudiado por parte de grupos independientes, consultoras de seguridad o foros universitarios, lo que origina en los primeros años de vida numerosos hallazgos en forma de fallos en la implementación o defectos a la hora de aplicar las especificaciones por parte de los fabricantes. Actualmente y pasados unos pocos años desde el despegue de la tecnología Bluetooth se han descubierto y documentado un buen número de vulnerabilidades y técnicas que permiten vulnerar ciertos dispositivos.

Palabras clave: seguridad bluetooth, bluesnarf, bluebug, helomoto, bluejackin

1. Introducción

El nombre de bluetooth viene del Rey danés Herald Blatnd, y la traducción literal del nombre significa "diente azul", este rey fue famoso por su habilidad para la comunicación y por iniciar la conversión al cristianismo del pueblo Vikingo. El logotipo esta formado por la composición de sus iniciales.

denominada Bluetooth Special Interest Group (SIG), cuyo objetivo fue diseñar una serie de protocolos y especificaciones destinadas a reemplazar tecnologías ya obsoletas.



* B
h b



Figura 1. Herald Blatnd y origen del logotipo.

Bluetooth es una tecnología desarrollada por una organización comercial fundada en 1998 y formada por empresas como Ericsson, Nokia, IBM, Toshiba, Microsoft e Intel entre otras,

Bluetooth está basado en tecnología de radio de bajo coste, y su uso está destinado a terminales móviles o estáticos cuyo alcance sea reducido. Funciona en una frecuencia de 2.45Ghz en redes "ad-hoc" denominadas "piconet" en las que pueden existir desde 2 hasta más de 200 clientes. En todas las redes piconet uno de los sistemas tiene que ejercer la función de "maestro". El maestro es definido por el sistema que inicia la conexión y es el encargado de establecer los parámetros iniciales. Existe la posibilidad de conectar varias piconet formando una "scatternet". En una scatternet un maestro no puede serlo de dos piconet, pero sí pueden ser esclavos de dos dispositivos de distintas piconet. El soporte de scatternet no es un requisito en el estándar de bluetooth y se puede implementar opcionalmente.

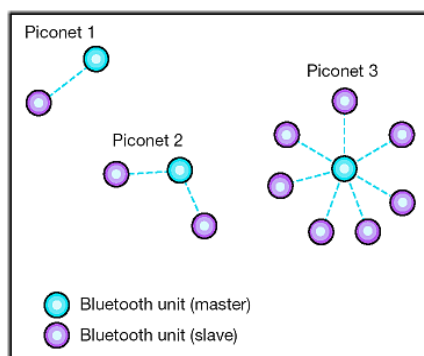


Figura 2. Ejemplo de Piconet.

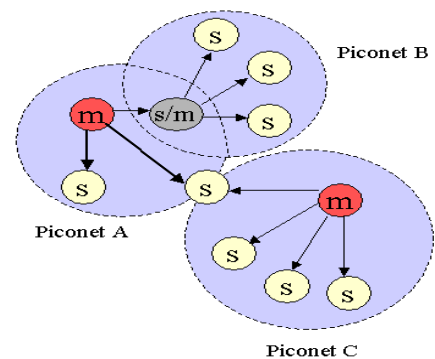


Figura 3. Ejemplo de Scatternet.

Bluetooth está concebido para sustituir el sistema de infrarrojos (IrDA), que tiene menos alcance y necesita que los dispositivos tengan visibilidad entre sí. En su última especificación denominada EDR (Enhanced Data Rate) ó 2.0 tiene un alcance máximo de 100 metros.

Cada uno de los dispositivos es identificado mediante su dirección BD_ADDR compuesta por 6bytes (similar a una dirección MAC en redes LAN) y ofrece distintos servicios a las unidades que acceden a él, especificados mediante el atributo "clase", un valor numérico de 24bits.

Como referencia pueden consultarse las especificaciones de este atributo en:

<https://www.bluetooth.org/foundry/assignnumb/document/baseband>.

La última versión del estándar de bluetooth es la 2.0, que se diferencia de su predecesora, la versión 1.2, en que mejor la tasa de transferencia, de 721Kb/sec a 2Mbit/sec, también reduce la duración del establecimiento del enlace, permite un mayor ahorro de energía y soluciona los problemas a la hora de convivir con otros dispositivos wireless, evitando interferencias con dispositivos 802.11b o microondas, mediante el uso de saltos de frecuencia.

1.2. Diferencias entre bluetooth y wireless

Bluetooth está diseñado para movilidad y economía, mientras que 802.11 está diseñado para sustituir a los cables en redes LAN. Por otra parte, 802.11 emplea más ancho de banda y recursos que bluetooth. Entre estas tecnologías existe la similitud de que ambas emiten en 2.4Ghz y pueden interferir una con la otra.

1.3. Seguridad.

La seguridad en bluetooth está diseñada para funcionar de tres formas distintas:

- Modo 1: permite conexiones desde cualquier dispositivo, como el utilizado en determinados perfiles PAN de algunos puntos de acceso.
- Modo 2: requiere una seguridad sencilla por servicio / aplicación a nivel L2CAP.
- Modo 3: utiliza procedimientos de seguridad antes de establecer el canal de la comunicación: uso de autenticación mediante PIN, filtro por dirección de origen (BD_ADDR) y cifrado mediante SAFER+.

1.4. Especificaciones.

Debido a que bluetooth está desarrollado para que este soportado por distintos productos y marcas muy dispersas, el protocolo tiene que ser idéntico en todos ellos.

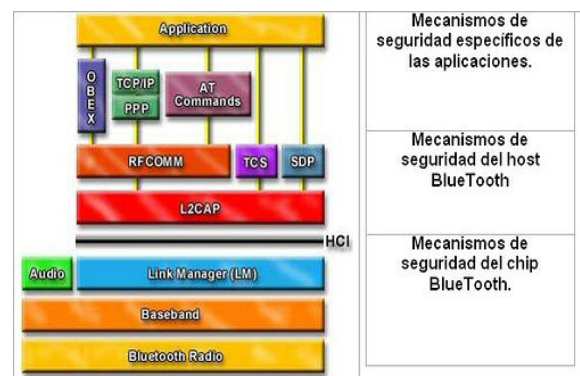


Figura 4. Pila bluetooth.

Las capas que lo componen son las siguientes:

- a) Bluetooth radio: se encarga de convertir los datos en señal de radio utilizando la modulación GFSK (Gaussian Frequency Shift Keying).
- b) Baseband: capa de la que depende el motor de bluetooth a la hora de construir paquetes, cifrar y descifrar, corrección de errores, mantener sincronización y otros parámetros de bajo nivel en la comunicación. Su comportamiento es administrado mediante instrucciones HCI.
- c) Link Manager: gestiona las comunicaciones, crea enlaces, comprueba su estado o los elimina si han terminado.
- d) L2CAP: gestionada mediante software, su función es determinar aspectos de alto nivel: quienes y donde están conectados, que requisitos tienen, si usan cifrado o no. Es la intermediaria entre la API y los protocolos más bajos.
- e) RFCOMM: es el protocolo encargado de emular un puerto serie para otros protocolos de capas superiores.
- f) TCS: gestiona las comunicaciones de audio.
- g) SDP: sistema por el que se consultan y sirven los servicios que ofrecen los dispositivos bluetooth, funciona como una aplicación cliente / servidor.
- h) OBEX: protocolo de intercambio de datos creado por IrDA.

1.5. Seguridad en autenticación.

Para que dos dispositivos creen un enlace en modo 3 es necesario que ambos conozcan una clave común denominada PIN. Estas claves generalmente son de cuatro dígitos, aunque la especificación actual admite hasta 16. Existe una debilidad en el intercambio de claves que permitiría a un atacante conseguir el PIN si es menor de 6 dígitos en menos de 12 segundos. Por otra parte, la gran mayoría de compañías que fabrican productos con un PIN establecido, asignan el mismo PIN a todas las unidades de ese producto, como por ejemplo "0000" a un sistema de manos libres.

El establecimiento de conexión denominado "paring" se realiza siguiendo el siguiente procedimiento: con el PIN, la longitud del PIN, la dirección BD_ADDR y un número aleatorio llamado IN_RANDOM, se genera un valor "Kinit". IN_RANDOM es transmitido en claro al segundo dispositivo, que junto con los mismos datos anteriores, creara su propio Kinit. Kinit es la clave utilizada para las siguientes transmisiones.

Los dispositivos generaran un nuevo valor aleatorio denominado LK_RANDOM y junto a BD_ADDR creará una nueva clave denominada LK_K. Ambos terminales realizarán intercambio de sus respectivos LK_RANDOM. Una vez los dos dispositivos tengan las dos claves: LK_Ka y LK_Kb, se realiza un XOR para obtener Kab. Si Kab es igual en ambos sistemas se concluye la autenticación. Kinit dejara de ser valida para el resto de comunicaciones.

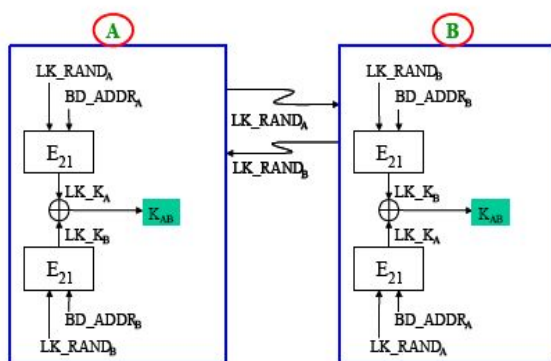


Figura 5. Creación de Kab.

Dependiendo de la negociación es posible que el sistema de paring sea distinto, usando uno más sencillo: se realiza un XOR de Ka con Kinit que da como resultado Klink, el dispositivo A, genera un valor aleatorio llamado AU_RANDa, este es

transmitido a B. Ambos dispositivos pueden generar SRES que esta compuesto de la siguiente forma: E1(AU_RANDa, BD_ADDR, Klink). Por último, se da por válida la autenticación si ambos dispositivos tienen el mismo valor SRES.

1.6. Cifrado de datos.

El uso de cifrado es opcional en la comunicación, y solo si se ha negociado, se generará una clave denominada Kc con el siguiente algoritmo: E3(AU_RANDa, Klink, COF). Donde AU_RANDa es un nuevo número aleatorio generado por el dispositivo A y transmitido a B, y COF variará en función del tipo de comunicación, pudiendo ser el desplazamiento del cifrado autenticado (ACO) o la concatenación de ambas BD_ADDR.

Los datos serán transmitidos utilizando un XOR de BD_ADDR, CLOCKa y Kc. El uso del valor CLOCKa (reloj del master) se realiza para dificultar el posible análisis de la trama cifrada.

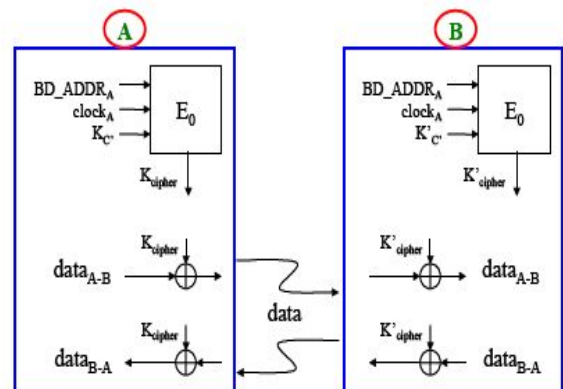


Figura 6. Transmisión cifrada.

2. Identificación de dispositivos.

El primer paso en un test de intrusión orientado a Bluetooth es identificar y localizar los dispositivos que vayamos a analizar, por tanto, la primera fase de la auditoria consiste en determinar los sistemas bluetooth accesibles.

2.1. Detección normal, análisis simple.

El tipo de análisis más simple que podemos realizar consiste en emplear las capacidades propias de los dispositivos bluetooth para localizar otros terminales que compartan el mismo espacio físico, lo que permite obtener sus direcciones BD_ADDR y sus nombres o alias asignados.

```
# hcitool scan
    22:22:22:22:22:22 PDA
    33:33:33:33:33:33 t630
    44:44:44:44:44:44 nokia660
```

Es muy probable que los nombres de los dispositivos muestren información útil sobre modelo y marca del sistema remoto.

2.2. Fingerprint mediante blueprint.

El grupo "Trifinite" mantiene una base de datos para realizar fingerprint de dispositivos, esta identificación se basa en que los 3 primeros bytes de la dirección BD_ADDR están asignados a una compañía y un hash de los servicios que ofrece cada uno de los terminales.

Ejemplo de ejecución para obtener el fingerprint:

```
# sdptool browse --tree
00:02:78:38:94:3C | ./bp.pl
    00:02:78:38:94:3C -nomac
    00:0A:D9@4063698
    device: Sony Ericsson T610
    version: R1L013
    date: n/a
    type: mobile phone
```

```
note: n/a
00:0E:07@4063698
device: SonyEricsson T630
version: n/a
date: n/a:
type: mobile phone
note: n/a
```

La base de datos está disponible en: http://trifinite.org/trifinite_stuff_blueprinting.html

2.3. RedFang.

Bluetooth soporta dos modos de funcionamiento: visible y no visible. Configurando el dispositivo como "no visible" se añade una nueva capa de seguridad permitiendo que nuestro terminal no sea detectado por terceros. Para descubrir estos dispositivos ocultos, se utiliza la herramienta "redfang", que realiza un ataque de fuerza bruta de direcciones BD_ADDR. Este ataque no se ejecuta a toda la dirección, con el fin de ahorrar tiempo, aun así, este proceso es muy lento y solo sirve como prueba de concepto. En la actualidad RedFang soporta múltiples mochilas (dongles) bluetooth en paralelo.

2.4. Bluespam.

Bluespam esta diseñado como prueba de concepto para realizar spam a teléfonos móviles y pdas que tienen determinados servicios abiertos.

A través del atributo "clase", se identifican que dispositivos son susceptibles de recibir mensajes de texto o imágenes. Ya que, por ejemplo, un sistema de manos libres no lo es.

Como prueba de concepto se puede utilizar el software disponible en:

<http://www.mulliner.org/palm/bluespam.php>

3. Vulnerabilidades.

A continuación se describen una serie de técnicas que permiten aprovechar fallos en las distintas implementaciones del protocolo.

3.1. Bluejacking.

Esta técnica consiste en enviar un contacto cuyo nombre es el texto que se desea que aparezca a la persona en la pantalla de su terminal. No implica ningún riesgo, exceptuando que se utilice para realizar ingeniería social o spam.

3.2 PSM Scanning.

Como se ha descrito anteriormente, SDP es el encargado de la publicación de los servicios disponibles, esta publicación varía en función de la implementación y configuración, pudiendo publicar todos los servicios disponibles o sólo algunos.

Existe la posibilidad de realizar una identificación de servicios mediante fuerza bruta, similar a lo que sería un análisis de puertos o "portscan" en TCP/IP.

Esta tarea se puede realizar con la herramienta BTAudit disponible en:

<http://www.betaversion.net/btdsd/download/>

3.3. Bluesnarf.

Bluesnarf es una de las vulnerabilidades más graves, aunque actualmente tiene un impacto muy bajo, puesto que solo afecta a terminales antiguos.

La transferencia de ficheros mediante bluetooth se realiza utilizando el protocolo OBEX, que permite varios métodos, como PULL o PUSH. Algunos terminales solo piden autenticación a la hora de realizar el enlace, cuando el método es PUSH, permitiendo obtener cualquier dato si se utiliza PULL.

Se muestra un ejemplo de la utilización de un ataque de BlueSnarf, con el cliente de Linux "obexftp", usando el servicio en el canal "10":

```
# obexftp -b 00:0F:11:11:11:11 -B 10 -g \  
telecom/pb.vcf
```

La siguiente lista muestran los archivos disponibles en algunos terminales móviles según Infrared Data Association (IrMC):

telecom/devinfo.txt: información del terminal
telecom/pb.vcf : agenda de teléfonos
telecom/rtc.txt: fecha y hora del reloj
telecom/note.vnt: notas
telecom/inmsg.vmg: mensajes recibidos
telecom/cal.vcs: calendario
telecom/sentmsg.vmg: mensajes enviados
telecom/outmsg.vmg: buzón de mensajes en salida

Se puede consultar la lista completa en:

http://new.remote-exploit.org/index.php/BT_main

Los dispositivos vulnerables se encuentran en la página: <http://www.bluestumbler.org>, entre ellos: Sony Ericsson T68, T68i, R520m, T610, Z1010 o Nokia 6310, 6310i, 8910, 8910i.

3.4. Bluesnarf++.

BlueSnarf++ es igual a bluesnarf pero permite la lectura y escritura de archivos (del mismo modo a como se haría en un FTP) en el sistema de ficheros virtual del terminal o de un dispositivo de almacenamiento extraíble (SD, MMC, etc) y listar sus ficheros. Actualmente no existe documentación detallada que explique su funcionamiento.

3.5. Bluebug.

Este fallo es considerado el más importante por su riesgo, consiste en acceder al terminal mediante RFCOMM utilizando un canal que SDP no pública, y ejecutar comandos AT. Con estos comandos se puede leer o mandar SMS, realizar llamadas o leer la agenda de un terminal móvil.

La lista completa de instrucciones AT está disponible en la web:

http://new.remote-exploit.org/index.php/BT_main

El siguiente ejemplo muestra una conexión con el servicio en el canal "17" que genera un dispositivo en el cliente, al cual se puede conectar con un terminal como minicom (9600 8N1):

```
# rfcomm connect 0 00:0F:DE:11:11:11 17
Connected          /dev/rfcomm0      to
00:0F:DE:11:11:11 on channel 17
```

Teléfonos vulnerables son por ejemplo : Sony Ericsson t610 o Nokia 6310i.

3.6. HeloMoto.

De reciente aparición HeloMoto consiste en una combinación de bluebug y bluesnarf. El atacante inicia una conversación con el objetivo enviado una vCard, esta conexión es interrumpida por el atacante. El objetivo considera que el dispositivo cliente es de confianza permitiéndole el uso de comandos AT en el servicio de manos libres. Esta vulnerabilidad afecta a teléfonos de la marca Motorola. Es posible obtener el código fuente de la aplicación que explota esta vulnerabilidad en la url: http://trifinite.org/trifinite_stuff_helomoto.html

3.7. Abuso del modo 3.

Consiste en crear un vínculo de confianza (trusted device) mediante ingeniería social, como por ejemplo mandando un contacto, y aprovechar este vínculo para conectar a otros servicios.

El problema reside en la falta de validación por servicios en vez de por dirección BD_ADDR.

3.8. BlueSmack.

Denegación de servicio al estilo "ping de la muerte" en la capa L2CAP. Este ataque se puede realizar con las herramientas de la pila Bluetooth de Linux: "bluez". El comando "l2echo", utilizado para realizar este DoS, permite especificar el tamaño del paquete con el parámetro "-s".

3.9. BlueBump.

Esta técnica toma el nombre de "bump keys", herramientas utilizadas para abrir cerraduras. El proceso consta de las siguientes fases:

- Entrar en la lista de dispositivos de confianza (mode-3-abuse, requiere de ingeniería social).
- Solicitar la eliminación de la clave del enlace sin perder la conexión.
- Volver a solicitar una nueva generación de clave para el enlace.

La correcta ejecución del proceso, permite permanecer conectado hasta que se elimine esta nueva clave.

3.10. BlueDump o Bluespoof.

Este ataque consiste en obtener el PIN o contraseña del enlace. Tiene como requisito conocer las BD_ADDR de los dispositivos que van a asociarse.

Se realiza una falsificación o "spoof" de una de las direcciones BD_ADDR y se envía una trama del tipo "HCI_Link_Key_Request_Negative_Reply". Cuando el objetivo de la falsificación solicite el PIN, en algunos terminales, este estado fuerza una renegociación del enlace.

4. Otras consideraciones.

4.1. Carwhisper.

Se denomina carwhisper a la realización de un enlace con el sistema de manos libres bluetooth de un coche. Esta técnica utiliza las contraseñas preestablecidas por los fabricantes.

Una vez establecido el enlace es posible escuchar conversaciones o intervenir en ellas.

Aunque es otro ejemplo de "usuarios por defecto", este método ha causado mucha repercusión mediática.

4.2. Bluetooth y Bluesniper.

Mediante esta técnica de hardware es posible modificar un dispositivo bluetooth para amplificar su señal y poder realizar ataques a mayor distancia. No tiene ninguna implicación a nivel de software. Se han conseguido establecer enlaces a una distancia de 1,1 millas o 1769 metros.



Figura 7. BlueSniper.

4.3. Gusanos.

Cabir es el primer gusano creado para sistemas móviles con Symbian que se transmite mediante bluetooth. Para que pueda realizarse la infección, es necesaria la intervención del usuario, aceptando la instalación de una nueva aplicación enviada por un tercero ya infectado. De este virus existen múltiples versiones con pequeñas modificaciones. Aunque su impacto es bajo, por ser el primer virus que se propaga a través de bluetooth resultó innovador.

5. Herramientas.

- Bloover: Symbian con MIDP2.
- Bloover II: No pública: Liberada en el congreso WhatTheHack.
- Bluenix: LiveCD No público Liberada en el congreso WhatTheHack.
- Bluepot: Honeygot de bluetooth. No pública.
- HeloMoto. Herramienta para explotar este fallo, del grupo Trifinite.
- Hcidump: Sniffer de capa HCI, incluida en el paquete de bluez.
- BlueSpam: cliente de Spam.
- BTClass: permite cambiar la “clase” de un dispositivo con PalmOS.
- BTAudit: realiza la comprobación de servicios que ofrece un dispositivo sin utilizar SDP.
- ObexFTP: cliente OBEX.
- bluez-utils: Herramientas básicas para la administración de bluetooth.

5. Conclusiones.

Por el momento, todos los fallos que se han publicado son específicos de dispositivos que realizan una incorrecta implementación del protocolo.

Por lo tanto, bluetooth se puede considerar seguro y son los desarrollos de los terminales los que no siguen adecuadamente el protocolo.

Se han encontrado vulnerabilidades en casi todos los productos de telefonía móvil y PDAs, así como en pilas de bluetooth de ordenadores personales, como la de Linux o la de MacOSX.

Medidas de seguridad.

- Desactivar Bluetooth si no se utiliza.
- Configurar el dispositivo como “no detectable”.
- Utilizar un nombre que no sea representativo de las especificaciones técnicas.
- Utilizar un PIN complejo.
- No aceptar conexiones de dispositivos desconocidos.
- Verificar, de forma periódica, los dispositivos tipificados como de confianza.
- Configurar el dispositivo para que acepte cifrado.
- Mantener el firmware actualizado.

Referencias

Para ampliar la información pueden consultarse las siguientes direcciones:

- <http://www.bluetooth.org>
- <http://www.trifinite.org/>
- <http://www.securityfocus.com/infocus/1836>
- http://www.blackops.cn/whitepapers/at_stake_smashing_teeth.pdf

- <http://www.geektown.de/index.php?catid=9&blogid=1>